2003P03862 B5

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 430 694 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
14.09.2005 Bulletin 2005/37

(21) Application number: **03747378.2**

(22) Date of filing: **09.04.2003**

(51) Int Cl.[7]: **H04L 29/06**, H04L 12/24, G06F 17/30

(86) International application number:
PCT/CA2003/000530

(87) International publication number:
WO 2003/094465 (13.11.2003 Gazette 2003/46)

(54) PATH OPTIMIZER FOR PEER TO PEER NETWORKS

WEGOPTIMIERER FÜR GLEICHRANGIGE NETZWERKE

DISPOSITIF D'OPTIMISATION DE CHEMIN DESTINE A DES RESEAUX D'HOMOLOGUES

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GR HU IE IT LI LU MC NL PT RO SE SI SK TR

(30) Priority: **06.05.2002 US 138336**

(43) Date of publication of application:
23.06.2004 Bulletin 2004/26

(73) Proprietor: **Sandvine Incorporated**
Waterloo, Ontario N2L 3V3 (CA)

(72) Inventor: **BOWMAN, Don**
Waterloo, Ontario N2L 5S7 (CA)

(74) Representative: **Frost, Alex John et al**
Boult Wade Tennant,
Verulam Gardens
70 Gray's Inn Road
London WC1X 8BT (GB)

(56) References cited:
WO-A-02/15035 **WO-A-02/31615**

- **MARC MORIN: "Managing P2P Traffic on DOCSIS Networks" SANDVINE, [Online] 13 February 2002 (2002-02-13), XP002247441 Retrieved from the Internet: <URL:www.sandvine.com> [retrieved on 2003-07-14]**
- **SANDVINE : "Sandvine to Present business Case at Investor Forum" PRESS RELEASE, [Online] 15 April 2002 (2002-04-15), XP002247440 Retrieved from the Internet: <URL:www.sandvine.com> [retrieved on 2003-07-14]**

EP 1 430 694 B1

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

[0001] The present invention relates to providing a peer to peer path optimizer (PPO). Such an optimizer may be used to examine peer to peer networking messages and dynamically and transparently redirect them to a cost efficient path.

BACKGROUND TO THE INVENTION

[0002] Peer to peer (P2P) networking has emerged as a popular form of exchanging data such as movies or music among individuals using the Internet. In a P2P network each computer in the network has the same responsibilities as each of the others, i.e. it is a "peer". Many variations of P2P networks have been created, at the time of writing the most prevalent being: Napster, Kazaa and Gnutella. The use of P2P for transferring large amounts of multimedia data such as movies or music has significantly increased the amount of information transmitted on the Internet.

[0003] P2P has led to increased financial pressure for network service providers. A network service provider is an entity that maintains a group of computers or nodes that form a network. Examples of networks include but are not limited to: a network controlled by an Internet Service Provider (ISP), a corporate network or a university network.

[0004] A network service provider typically must pay a fee for the traffic to and from their network.

[0005] Given the popularity of P2P networking, it is difficult for any network service provider to block P2P traffic. The network service provider is left with few choices, namely:

a) tiered bandwidth services, and the hope that users will pay for additional bandwidth, or

b) capping the amount of bandwidth available to P2P applications, which could cause dissatisfaction among the user base.

[0006] Thus, there is a need for an alternative approach, which allows a network service provider to cost effectively constrain P2P traffic through their network, while maintaining or improving existing performance to the user. The present invention addresses this need.

[0007] WO 02/15035 discloses an arrangement for intelligently directing a search of a peer to peer network, in which a user performing a search is assisted in choosing a host which is likely to return fast, favourable results to the user.

SUMMARY OF THE INVENTION

[0008] The present invention is directed to a peer to peer optimizer as set out in claim 1. The optimizer examines peer to peer messages between nodes within networks connected to the optimizer, for the purpose of optimizing behavior on each of said networks, and in particular, for the purpose of determining a cost efficient path for each peer to peer message.

[0009] The present invention is also directed to a corresponding process and a corresponding computer readable medium.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] For a better understanding of the present invention, and to show more clearly how it can be carried into effect, reference will now be made, by way of example only, to the accompanying drawings in which:

Figure 1 is a block diagram of example networks suitable for connecting to a peer to peer optimizer of the present invention;

Figure 2 is a block diagram of a plurality of nodes connected to a PPO;

Figure 3 is a block diagram of an ISP system;

Figure 4 is a block diagram of a centralized server network;

Figure 5 is a block diagram of a centralized server network utilizing a PPO;

Figure 6 is a block diagram of a decentralized server network;

Figure 7 is a block diagram of a decentralized P2P network utilizing a PPO;

Figure 8 is a block diagram of a hybrid P2P network;

Figure 9 is a block diagram of a hybrid P2P network utilizing a PPO;

Figure 10 is a block diagram of a PPO;

Figure 11 is a logical flow diagram illustrating the processing of a ping message;

Figure 12 is a logical flow diagram illustrating the processing of a pong message;

Figure 13 is a logical flow diagram illustrating the processing of a query message;

Figure 14 is a logical flow diagram illustrating the processing of a queryhit message; and

Figure 15 is a logical flow diagram illustrating the processing of a connect request.

DETAILED DESCRIPTION OF THE INVENTION

[0011] Figure 1 is a block diagram of networks connected to a peer to peer optimizer of the present invention. Peer to Peer optimizer (PPO) 10 monitors all P2P traffic between a plurality of networks 12. Examples of networks 12 include but are not restricted to; a network controlled by an ISP, a corporate network, or a University network. Networks 12 would typically be connected to PPO 10 via the Internet, but that is not a requirement of the present invention. Any network 12 that is capable of providing or requesting P2P traffic may make use of PPO 10.

[0012] To minimize the cost of P2P traffic, network 12 utilizes PPO 10 to determine a cost efficient path for exchanging P2P data between nodes 14. A node 14 is any computer that is capable of receiving or transmitting P2P data.

[0013] Referring now to Figure 2, a block diagram of a plurality of nodes connected to a PPO 10 is shown. Each network 12a and 12b contains a plurality of nodes 14. For each node 14 that it is aware of, PPO 10 maintains a cost class. Table 1 illustrates the cost class for each node of Figure 2.

Table 1.

Node	Cost Class
14a	5
14b	5
14c	10
14d	250
14e	150
14f	50
14g	250

[0014] Assuming that a P2P request can be serviced within a single network such as 12a, then typically the most cost efficient paths for P2P transfer will be within network 12a. Examples would be connections to nodes 14a and 14b. However, this may not always be the case. For example a request to node 14d may be very expensive if node 14d which contains the data, resides halfway around the world within a corporate intranet. In such a scenario, node 14f, within network 12b, which contains the required data, would be a more cost efficient choice.

[0015] In determining a cost efficient path for the delivery or reception of P2P data, PPO 10 combines the cost class of each node on the end of a potential exchange of data. This combination results in a path cost value. For example, a request from node 14e for a file on node 14a may result in a path cost of 155. This example is one of simple addition to the cost class of two nodes to determine a path cost. The inventor does not intend to restrict the present invention to any specific algorithm to obtain a path cost. For example, a weighting factor may be applied to nodes with a high cost class to exclude them from consideration in calculating a path cost.

[0016] In Figure 3, a block diagram of an ISP system is shown. Figure 3 illustrates how an Internet Service Provider

(ISP) may make use of the present invention. Network 12a is the network maintained by an ISP. Network 12a is connected to a plurality of networks 12b to 12n via links 30b to 30n. Typically networks 12b to 12n would be accessible via the Internet, but they may be any form of network which contains files for P2P exchange. Network 12a comprises a plurality of nodes 14. P2P data is exchanged between nodes 14 within network 12a and nodes within networks 12b to 12n. A node 14 may be the computer of a home user, a business computer or a corporate server connected to any of networks 12a to 12n. Returning to the present ISP example of Figure 3, each node 14 within network 12a is connected to a communications module 20, which allows node 14 to communicate with network 12a. Communications module 20 may be Digital Subscriber Line Access Multiplexer (DSLAM) which is used for phone line connections. Communications module 20 may also be a Cable Modem Termination System (CMTS) which is used for cable connections. Communications module 20 may also be a module that accepts dialup connections, wireless connections or fibre optic connections. The point here being that communications module 20 connects a node 14 to network 12a. An aggregator 22 collects the data to and from communications modules 20 and is connected to distribution router 24 for this purpose. Distribution router 24 determines where a request for information should be routed within network 12a. Distribution router 24 is connected to core router 26, one or more cache servers 28 and one or more P2P Path Optimizers (PPO) 10. Core router 26 is connected to networks 12b to 12n most typically by interXchange Carrier (IXC) links 30b to 30n. An IXC is a telecommunications company such as AT&T. A cache server 28 is a repository of information obtained from networks 12b to 12n that may be frequently accessed by nodes 14. To avoid the expense of continually requesting data from networks 12b to 12n network 12a may store frequently accessed information in one or more caches 28. Most commonly this would be current versions of popular websites, but may include all forms of data. Cache 28 may also commonly be referred to as a "cache cluster" or a "cache server".

[0017] PPO 10 is where the present invention resides. PPO 10 serves to provide three main functions:

1) Reorganize networks connected to PPO 10. This is achieved by intercepting all P2P messages and attempting to have nodes connect to other nodes in the same cost class. This allows the networks to reorganize in two ways:

   a) they become flatter as nodes connect to nodes under the control of a PPO 10 thus a tree of connections between nodes would have at most a depth of one; and
   b) PPO 10 attempts to connect nodes to other nodes within a network, where without the use of a PPO 10, connections would be random and a tree of connections between nodes may have an unlimited depth.

2) Reduce network traffic. This is done by not broadcasting messages but instead sending them where they need to go, or dropping them if there is no need to send them on.

3) Redirecting traffic to a cost efficient path.

Each of these functions is discussed in more detail below.

[0018] Although the example of Figure 3 applies to the network of an ISP, it is not the intent for the inventors to restrict the use of the present invention to an ISP network. Any network for which an entity wishes to control P2P traffic in a cost efficient manner may make use of the present invention. As discussed above examples would be a corporate network or a University network or any commercial use of a large network, such as the hotel industry.

[0019] Before describing in detail the structure of PPO 10, we will refer first to how it may be utilized in a variety of P2P models.

[0020] Referring now to Figure 4, a block diagram of a centralized server network is shown generally as 40. Network 40 is an example of a first generation P2P network, such as Napster. Napster is an Internet service that was originally designed to permit users to exchange MP3 music files. In network 40, a central server 42 connects a plurality of nodes 14 via connections 44. Connections 44 would typically be connections via the Internet. A node 14 sends a request for a file to central server 42 via a connection 44. Central server 42 provides a reply via connection 44 indicating on which node 14 the requested file resides. In essence central server 42 contains a directory of all files available for access on all nodes 14. In acting on the reply, the requesting node 14 establishes a connection with its peer node 14 that contains the file and requests a copy of the file, as is shown in transfer link 46.

[0021] To explain how Figure 4 may make use of the present invention we refer now to Figure 5 where a block diagram of a centralized server network utilizing a PPO is shown generally as 50. In network 50, PPO 10 examines search requests sent to central server 42. If PPO 10 is aware of the requested file, it will provide the requester with a cost efficient path to the file. If PPO 10 is not aware of the file it will utilize alternatives to direct the requestor to the file. These alternatives are discussed in detail later. In determining which node 14 to direct the request to, PPO 10 makes use of cost class information. Cost class information for a node would typically be determined by metrics such as the speed of the connection to the node (e.g. bandwidth and distance) and the monetary cost of using a connection to the node. Cost class information would typically reflect a monetary cost to obtain a file from a specific node 14. An

administrator of the present invention may set their own cost class to a node 14 or PPO 10 may set them by default or determine them dynamically. Whatever the method of establishing the cost class for a node, the point is that each node has a cost associated with it and PPO 10 utilizes this information to provide a cost efficient path for exchanging P2P data.

[0022] Referring to Figure 6, a block diagram of a decentralized server network is shown generally as 60. Network 60 utilizes a distributed model where each node 14 is equal and there is no central server 42 as with network 40 (Figure 4). Network 60 may be considered to be a second generation P2P network, an example of which would be Gnutella. Gnutella provides a file sharing service for many types of information and is not directed solely to the exchange of multimedia files. Each node 14 tries to maintain some number of connections 44 to other nodes 14 at all times. Requests for information are sent with a Time to Live (TTL) field that is decremented and then forwarded by each node 14 to all other nodes 14 to which it is connected. When the TTL value reaches zero, the request is dropped. This type of network has been shown to have significant scaling issues, as requests for information will degrade network performance. As with network 40 when a requested file is located a direct connection is made between two nodes to transfer the requested data as shown by transfer link 46.

[0023] In the present invention the topology of network 60 is reconfigured as shown in Figure 7. Figure 7 is a block diagram of a decentralized P2P network utilizing PPO 10 and is shown generally as 70. In network 70 when P2P communication is sent between networks 12a and 12b PPO 10 examines it. PPO 10 then determines a cost efficient manner to deal with the communication. It is not the intent of the inventor to restrict the use of the present invention to only two networks 12a and 12b as shown in Figure 7.

[0024] Referring now to Figure 8 a block diagram of a hybrid P2P network is shown generally as 80. This network topology may be referred to as third generation P2P where some nodes are elected as "supernodes" or "ultra peers" and serve as the traffic coordinator for the other nodes. In Figure 8, supemodes are designated with the feature number 82 and are connected to each other. This model is utilized by P2P services such as Fasttrack, Kazaa, Morpheus and Grokster. The supemodes 82 change dynamically as bandwidth and network topology change. Any node 14 may be a supemode 82.

[0025] Referring now to Figure 9, a block diagram of a hybrid P2P network utilizing PPO 10 is shown generally as 90. In network 90, PPO 10 acts as a supernode between networks 12a and 12b. All nodes within network 12a will see PPO 10 as their supemode and thus as their path to network 12b. Nodes 14 within network 12a may also be supemodes within network 12a (not shown).

[0026] With regard to the topologies of the networks shown in Figures 4 to 9, it is the intent of the inventor to simply illustrate how the present invention may be utilized in existing P2P networks. It is not the intent of the inventor to restrict the present invention to the networks shown, but rather to provide examples of the diversity of the present invention.

[0027] Referring now to Figure 10 a block diagram of a PPO is shown generally as 10. As one skilled in the art can appreciate, PPO 10 may be implemented in many different ways. The structure of PPO 10 as shown in Figure 10 serves only as an example of one implementation that may be used to examine and manage P2P communications. We will now describe the components of PPO 10 as illustrated in Figure 10 in more detail.

[0028] Licensing module 102 is responsible for enforcing the maximum number of concurrent users of PPO 10 for which the customer (i.e. the owner of a PPO 10) has paid a license fee. Configuration module 104 maintains the configuration of PPO 10, such as the sub-networks and IP addresses of the nodes that reside within a network 12. Statistics module 106 maintains the statistics for PPO 10, such as the number of files redirected and the number of concurrent users. Logging module 108 is responsible for logging functions, such as when PPO 10 was started up or shut down and when the number of licenses was exceeded. Load balancer feedback module 110 provides a negative feedback loop to an external load balancer so that multiple PPO's under the control of a customer will receive equal traffic. WCCP module 112 operates with the Cisco Web Cache Communication Protocol (WCCP) to ensure that a router, such as distribution router 24 of Figure 3 sends only P2P communications to one or more PPO's 10. As one skilled in the art can appreciate, a number of methods may be used to direct P2P traffic to a PPO 10, such as recognizing specific port addresses or context sensitive scanning of packets. WCCP serves only as one example. GUID generator 114 generates a globally unique identifier for each sender of a P2P packet to avoid the possibility of looping back to the original sender of the packet and to also uniquely identify messages that have been received.

[0029] P2P application 116 acts as the control program for PPO 10. Application 116 comprises: route/path cost module 118, query module 120, ping/pong network training module 122, connection manager module 124 and transfer manager module 126. Route/path cost module 118 assigns a path cost to each proposed connection based upon the cost class of each node in the connection.

[0030] Query module 120 comprises: string edit distance module 128, search amalgamation module 130, query routing logic module 132, QoS modification module 134 and content index module 136. String edit distance module 128 determines the similarity between the name of a requested file and the filenames known to PPO 10. Search amalgamation module 130 utilizes string edit distance module 128 to map the name of a requested file to the known files available, regardless of cost class. Query routing logic module 132 routes queries for a file to the nodes that are

likely to contain the requested file. Module 132 maintains a list of all messages to and from a network 12. By maintaining such a list, module 132 may quickly drop spurious messages, such as requests for data that have not been acknowledged. QoS modification module 134 rewrites the routing information of module 132 to select a cost efficient path determined by route/path cost module 118. Routing information includes QoS parameters such as stated bandwidth and uptime. The purpose of rewriting routing information is to provide the requestor with a path to a file or files that make the most efficient use of network resources. By doing so a message may be redirected. Content index 136 maintains an index of content available for access in nodes 14 within networks 12. Content index 136 also contains the cost class for each node in which the content resides. Typically such content will be a file but may also include forms of data such as streaming media. It is not the intent of the inventor to restrict the use of the term "file" to any form of P2P data that may be examined by or transmitted through PPO 10.

[0031] Ping/Pong network training module 122 serves to fill host cache 138 with IP addresses of nodes 14 based upon the Ping messages received by PPO 10 from nodes 14. Ping/Pong network training module 122 sends a plurality of Pong messages in response to a Ping message in an attempt to train a network sending a Ping message. Pong messages are sent by PPO 10 for each node 14 that is in the same cost class as the sender of the Ping that PPO 10 is aware of. This use of multiple Pong messages serves to train the network that sent the Ping. This training provides the sending network with nodes other than those for which PPO 10 wishes to restrict traffic.

[0032] When a connection is established between a node 14 and PPO 10, connection manager 124 maintains the connection until the node 14 drops the connection. Index fetch module 142 is responsible for obtaining content names and adding them to content index 136.

[0033] Transfer manager 126 is in essence a proxy that handles the exchange of P2P data. Manager 126 utilizes fetch redirection module 144 to redirect a request for content to a node with a lower path cost. A node 14 may make a request for a specific file on another node 14. If that file is available via a more cost efficient path, fetch redirection module 144 will silently direct the request to another node having a more cost efficient path.

[0034] A plurality of P2P protocol specific handlers 146 are responsible for maintaining a specific P2P protocol, for example Gnutella or Fasttrack. Transmission Control Protocol (TCP) handler 148 ensures the maintenance of correct TCP behaviour. Similarly, Internet Protocol (IP) handler 150 serves the same purpose for IP. It is not the intent of the inventor to restrict the present invention to the use of TCP and IP. These serve only as an example. As one skilled in the art can appreciate any number of communication protocols may be used, including, but not restricted to: ATM, UDP, and wireless.

[0035] Differentiated Services Code Point (DSCP) marking module 152, utilizes Differentiated Services (DiffServ or DS) to specify IP packets by class so that certain types of packets get precedence over others. For example a limit may be imposed on the number of P2P packets allowed to enter or leave a network 12. Such a feature is optional but may be used by networks that find P2P data is consuming too much of their bandwidth. As one skilled in the art can appreciate any number of schemes such as packet snooping or recognizing specific port addresses may be utilized to identify P2P traffic. It is not the intent of the inventor to restrict the ability to limit P2P traffic to the DSCP solution.

[0036] PPO 10 optimizes behavior between and within the networks 12 to which it is connected. Behavior is the ability to create, destroy, modify or ignore messages. Behavior optimizes future behavior of each network 12, not just the current message. An example of creating a message is a false pong. An example of destroying a message is deleting a message that has already been answered or in the case of Gnutella, a message whose TTL has expired. Modification is not limited to QoS modification module 134. For example, search amalgamation module 130 may modify messages to reflect the closest filename as determined by string edit distance module 128. In the case of a specific protocol, for example Gnutella, modification may include overwriting the TTL portion of the message when forwarding the message. Similarly the GUID for a message may be changed if needed. In essence, depending upon the protocol, PPO 10 may modify messages as required to optimize network behavior. An example of ignoring a message is to ignore a query request to a node in a network, as traffic from that network has been restricted.

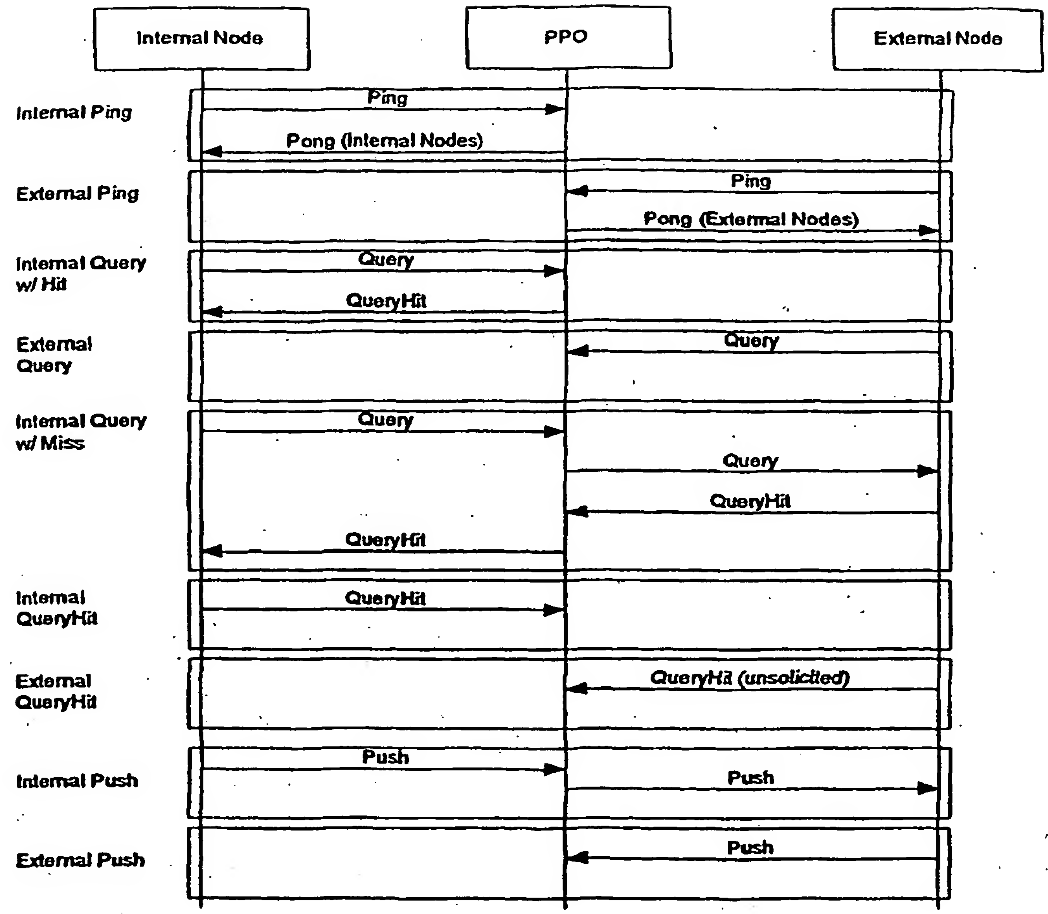
[0037] In order for PPO 10 to examine and act upon P2P requests, it must be aware of a variety of P2P protocols. This functionality is handled by P2P protocol specific handlers 146.

[0038] By way of example we refer next to how a P2P protocol specific handler 146 may interface with the Gnutella protocol. It is not the intent of the inventor to restrict the present invention to work simply with the Gnutella protocol, but rather to provide a practical example of how the present invention may deal with P2P requests.

[0039] The Gnutella protocol has five message types, namely: ping, pong, query, queryhit and push. How a handler 146 handles each of these messages is shown in the following Chart 1. In Chart 1, the term "internal node" refers to a node 14 within network 12a of the ISP example of Figure 3. The term "external node" refers to a node 14 within a network 12b to 12n of Figure 3. By the use of the terms "internal node" and "external node" the inventor means to show how PPO 10 may be used to examine and redirect P2P traffic between nodes in an "internal" network such as an ISP and nodes in an "external" network such as a plurality of sites on the Internet.

[0040] Chart 1 illustrates how PPO 10 may be used to examine and redirect P2P traffic between network. In the following description of Chart 1, we advise the reader refer to Figures 3 and 10 as well as Chart 1.

Chart 1



[0041] A ping message is used to determine if a node 14 is active, and helps to establish a database of active nodes in host cache 138 of Figure 10. PPO 10 responds to a ping message with a pong message. To avoid identifying a node 14 within network 12a, PPO 10 would typically provide a forged pong message. The forged pong message would indicate the number of files shared and the amount of data shared within network 12a containing the pinged node 14, as well as the IP address and port of the pinged node. Similarly in this example, PPO 10 does not forward pong messages, however it does receive them and adds them to the list of nodes in host cache 138 from which it may obtain data.

[0042] A query message is a search message containing a fragment of a filename, in other words, a request for data. In the present example, incoming query messages from an external node are dropped, thus appearing to be a query miss and thereby avoiding servicing a P2P request from a network 12b to 12n. It is not the intent of the inventor to require that query messages be dropped, it is simply one method that may be used to restrict unwanted P2P traffic into network 12a. Implementations utilizing PPO 10 may choose to allow free flow of all messages or to provide a limited amount of traffic. Query messages from a node 14 within network 12a are forwarded first to the nodes 14 containing the requested file that have a cost efficient path. Typically these would be nodes 14 within network 12a, but that may not always be the case. The nodes 14 having the requested data will then respond with queryhit messages. If there are no matches for the request for data, or if no queryhit message is returned, then the query message is sent to a random set of nodes 14 within network 12a. One method of determining the random set of nodes 14 to receive the query message would be to use a weighted probabilistic function such as a round robin method based upon the number of files available from each node 14. In this way, the query does not always go to the node 14 having the largest

number of files. If there is still no match, the query is forwarded to nodes 14 having the lowest path cost in networks 12b to 12n.

[0043] A queryhit message is a response to a query message. Incoming queryhit messages from nodes in networks 12b to 12n are forwarded to the appropriate node 14 within network 12a. Incoming queryhit messages from nodes 14 within network 12a are forwarded back to the requesting node within network 12a and not sent out to networks 12b to 12n.

[0044] A push message is used when the transmitting node has a firewall and the receiving node does not. The receiving node sends a push message, which causes the transmitting node to open a connection directly to the receiving node. Incoming push requests may be optionally dropped by PPO 10 and are propagated unchanged on the way out of network 12a.

[0045] By way of example on how the present invention may be utilized to provide support for the Gnutella protocol, we will now refer to logical flow diagrams 11 to 15. As with the previous discussion with regard to Chart 1, we will be referring to the components of Figure 3 by way of example.

[0046] Referring now to Figure 11, a logical flow diagram illustrating the processing of a ping message is shown generally as 160. Beginning at step 162, a ping message is received. At step 164 the ping message is optionally dropped and if dropped, is not propagated within network 12a. At step 166 a forged Pong message is created. The forged pong response may contain the number of files available for P2P exchange within network 12a. The forged pong message may be sent to each node to connected to PPO 10 in order to train a network as described earlier with reference to ping/pong training module 122 of Figure 10.

[0047] Referring now to Figure 12, a logical flow diagram illustrating the processing of a pong message is shown generally as 190. Process 190 begins at step 192 with the receipt of a pong message. At step 194 a test is made to determine if the message is from a node 14 within network 12a. If the message is from a node 14 within network 12a the message may be optionally dropped at step 196. If the message is from a node 14 in a network 12b to 12n, processing moves to step 198 where the TTL for the message is decremented. Processing then moves to step 200 where a test is made to determine the current value of the TTL. If the TTL has expired, the message is dropped at step 202. If the TTL has not expired, processing moves to step 204 where a test is made to determine if a ping message to match the pong message has been received. PPO 10 stores messages it receives under the control of query routing logic module 132 (Figure 10). Typically a message is not stored for long as most P2P requests for information are resolved within less than a minute. If no matching ping message is found, the pong message is dropped at step 202. If a matching ping message has been seen, then the pong message is forwarded to the source of the original ping message at step 206.

[0048] In the above description of Figure 12, the inventor makes reference to Time To Life (TTL). TTL information is utilized by the Gnutella protocol, but not by all other protocols. For other protocols not recognizing TTL, the logic if Figure 11 would be modified to remove steps 198, and 200. Thus control would flow from step 194 in the negative case directly to step 204.

[0049] Referring now to Figure 13, a logical flow diagram illustrating the processing of a query message is shown generally as 210. Process 210 begins at step 212 where a query message is received. At step 214 a test is made to determine if the query message came from a node 14 within network 12a. If this is the case processing moves to step 216 where a test is made to determine if the requested file is contained within network 12a as indicated by content index 136 (Figure 10). If the file is contained within network 12a then processing moves to step 218 where the query message is forwarded to nodes 14 having the lowest cost class within network 12a. If the file is not found within network 12a then processing moves to step 220 where the query message is forwarded to a select weighted list of nodes 14 within network 12a. The intent here being that content index 136 may not be current and the requested file may reside within network 12a. One method of determining the set of nodes 14 to send the query message to would be to use a weighted probabilistic function such as a round robin method based upon the number of files available from each node 14 within network 12a. In this way, the query does not always go to a node 14 having the largest number of files.

[0050] A test is next made at step 222 to determine if the file has been located on a node 14 within network 12a. If the file has been located the location information is forwarded to the originator of the query message at step 224. If at step 222 the file has not been located, the query message is forwarded to a weighted subset of connected nodes having the lowest cost class in networks 12b to 12n at step 226. As mentioned before, a weighted round robin scheme may be utilized to select the nodes 14 in networks 12b to 12n to receive the query. A connected node is one that has established a communication path with PPO 10, for example via TCP/IP. Returning to step 214 if the query message is not from a node 14 within network 12a, processing moves to step 228 where the TTL value of the message is decremented. A test is then made at step 230 to determine if the TTL value for the message is greater than zero. If it is not, then the message is dropped at step 232 and processing ends. If the TTL value is less than or equal to zero then processing moves to step 226 where the query message is forwarded to all connected nodes in networks 12b to 12n. Optionally, if the query is from a node in networks 12b to 12n, the query may simply be dropped or returned to the requesting node at step 226, thus not requiring PPO 10 to forward the query to connected nodes.

[0051] As discussed above with reference to Figure 12, if the communication protocol does not make use of TTL, then steps 228, 230 and 232 would be deleted. The negative case from step 214 would then flow to step 226.

[0052] Referring now to Figure 14, a logical flow diagram illustrating the processing of a queryhit message is shown generally as 240. Process 240 begins at step 242 where a queryhit message is received by PPO 10. Processing moves to step 244 where the TTL for the queryhit message is decremented. At step 246 if the TTL is less than or equal to zero than the message is dropped at step 248. If the TTL is greater than zero, processing moves to step 250 where a test is performed to determine if a matching query message had been received for the queryhit message. PPO 10 stores messages it receives under the control of query routing logic module 132 (Figure 10). Typically a message is not stored for long as most P2P requests for information are resolved within less than a minute. If no matching query message was received, processing moves to step 248 where the message is dropped. If a matching query message was received, processing moves to step 252 where a test is made to determine if the queryhit message was from a node 14 within network 12a. If not, the queryhit message is then optionally forwarded to the node that made the original query at step 254. If at step 252 the queryhit message is determined to have come from a node 14 within network 12a, then processing moves to step 256. At step 256 a test is made to determine if the original query message corresponding to the queryhit message was from a node 14 within network 12a. If so, processing moves to step 254 where the message is forwarded to the node that made the original query. If not, processing moves to step 248 where the message is dropped.

[0053] Referring now to Figure 15 a logical flow diagram illustrating the processing of a connect request is shown generally as 260. Any node 14 may request a connection with any other node 14 at step 262. At step 264 a test is made to determine if the request is from a node 14 within network 12a. If so, connection manager 124 (see Figure 10) attempts to service the query through query module 120 (see Figure 10) to determine a cost efficient path within network 12a, at step 266. If at step 264 it is determined that the connect request is not from a node 14 within network 12a, processing moves to step 268. At step 268 a ping message is sent to connected nodes 14 within networks 12b to 12n. At step 270 one or more pongs are received and a decision is made at step 272 which connection to a specific external node 14 should be utilized. Step 272 may utilize a variety of methods to determine which connections to keep and which to drop. Typically, step 272 would maintain connections based upon the amount of data, cost class, and the total number of connections that may be maintained. If at step 272 a node 14 is found to be no better than an existing connection, it is dropped at step 274. If at step 272 a better connection is found, it is added to content index 136 at step 276.

[0054] Although this disclosure and the claims appended hereto make use of the terms query, queryhit, ping, pong, push and connect, it is not the intent of the inventor for these terms to be specifically associated with the Gnutella protocol. To the inventor the term query is analogous to a request for data and queryhit to a reply to a query, indicating that the data has been located. A ping is a standard computer communications term and is short for Packet Internet Groper; in essence it is a message to determine whether a specific computer in a network is accessible. A pong is a response to a ping. A push is a message sent directly to a node that is protected by a firewall. A push is used to request a direct connection between the node behind the firewall and the node sending the push message so that the node behind the firewall can "push" data to the requesting node. A connect is a connection between two nodes.

[0055] Although the disclosure refers to a PPO within an ISP by way of example, it is not the intent of the inventor to restrict the invention to such a configuration. For example a PPO may be used within any network, including networks utilized by corporations to exchange data with their employees or customers. Further, multiple PPO's may be utilized to provide redundancy in case one PPO fails and also to provide load balancing. In the case of a network 12 utilizing a single PPO, if the PPO failed, network 12 would revert to the status quo without the PPO; i.e. all P2P messages are exchanged with no decision made on who should service the request.

[0056] Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the scope of the invention as outlined in the claims appended hereto.

Claims

1. A peer to peer traffic optimizer (10), said optimizer comprising:

means for monitoring all peer to peer messages between and within networks (12) connected to said optimizer;
means for maintaining a cost class information of each peer to peer node (14);
means for assigning a path cost to each proposed connection based upon the cost class of each node (14) in the connection;
means for utilizing said path cost to determine a cost efficient path for said proposed connection; and
means for creating, destroying, modifying or ignoring peer to peer messages such that said proposed con-

nection uses said determined cost efficient path.

2. The optimizer of claim 1 wherein if a peer to peer message is not initially directed to said cost efficient path, redirecting said peer to peer message to said cost efficient path.

3. The optimizer of claim 1 wherein if a peer to peer message is not from a node in a first network and is a query message, forwarding said query message to one or more nodes not in said first network.

4. The optimizer of claim 1 wherein if a peer to peer message is from a node in a first network and is a query message and if said query message is unsuccessful, forwarding said query message to a weighted list of nodes in said first network.

5. The optimizer of claim 1 wherein if a peer to peer message is not from a node in a first network and is a query message, ignoring said query message.

6. The optimizer of claim 1 wherein if a peer to peer message is a queryhit message, determining if a matching query message has been observed and if so forwarding said queryhit message to the originator of said matching query message.

7. The optimizer of claim 1 wherein if a peer to peer message is a query message said optimizer determining:

a) if said query is from a node in a first network, determining if said query may be serviced by one or more nodes in said first network, if so, forwarding said query to said one or more nodes in said first network; or

b) if said query is from a node not in said first network, dropping said query.

8. The optimizer of claim 1 wherein if a peer to peer message is a pong and a matching ping message has not been seen, dropping said pong.

9. The optimizer of claim 1 wherein if a peer to peer message is a ping, said optimizer optionally dropping said ping.

10. The optimizer of claim 9 wherein if said optimizer has not dropped said ping, said optimizer creating one or more pong messages to nodes in the same cost class as the node sending said ping.

11. The optimizer of claim 1 wherein if a peer to peer message is a connect, and if said connect is better than an existing connection, dropping said existing connection for said connect.

12. The optimizer of claim 1 wherein said cost class is set by an administrator of said optimizer.

13. The optimizer of claim 1 wherein said cost class is set by said optimizer.

14. The optimizer of claim 1 wherein each peer to peer message is examined to obtain information about the originator of said message.

15. The optimizer of claim 14 wherein said information is stored by said optimizer in a host cache.

16. The optimizer of claim 1 further comprising a content index, said content index comprising a mapping of content to nodes.

17. The optimizer of claim 16 wherein said content index is utilized to limit the number of nodes a query message is forwarded to.

18. The optimizer of claim 17 wherein said content index is utilized to determine an optimized set of nodes to receive said query message, set optimized set of nodes based upon path cost.

19. The optimizer of claim 18 wherein if no nodes are found in said content index to satisfy said query message, forwarding said query message to a set of nodes determined by a weighted probabilistic function.

20. The optimizer of claim 1 wherein the means for creating, destroying, modifying or ignoring, comprises means for creating.

21. The optimizer of claim 1 wherein the means for creating, destroying, modifying or ignoring, comprises means for destroying.

22. The optimizer of claim 1 wherein the means for creating, destroying, modifying or ignoring, comprises means for modifying.

23. The optimizer of claim 1 wherein the means for creating, destroying, modifying or ignoring, comprises means for ignoring.

24. A peer to peer traffic optimizing process (10), said process comprising the steps of:

monitoring all peer to peer messages between and within networks (12);
maintaining a cost class information of each peer to peer node (14);
assigning a path cost to each proposed connection based upon said cost class of each node (14) in the connection;
utilizing said path cost to determine a cost efficient path for said proposed connection; and
creating, destroying, modifying or ignoring peer to peer messages such that said proposed connection uses said determined path.

25. The process of claim 24 further comprising the step that if a peer to peer message is not initially directed to said cost efficient path, redirecting said peer to peer message to said cost efficient path.

26. The process of claim 24 further comprising the step of utilizing a message to obtain information about the originator of said message.

27. The process of claim 26 further comprising the step of storing said information in a host cache.

28. The process of claim 24 further comprising the step of matching a requested filename to filenames stored in a content index.

29. The process of claim 28 further comprising the step of utilizing said content index in determining said cost efficient path.

30. The process of claim 24 wherein the step of creating, destroying, modifying or ignoring comprises modifying.

31. The process of claim 24 further comprising the step of forwarding a peer to peer message.

32. The process of claim 24 wherein the step of creating, destroying, modifying or ignoring comprises ignoring.

33. The process of claim 24 wherein the step of creating, destroying, modifying or ignoring, comprises creating.

34. The process of claim 24 wherein the step of creating, destroying, modifying or ignoring, comprises destroying.

35. The process of claim 24 further comprising the step that if a peer to peer message is not from a node in a first network and is a query message, forwarding said query message to one or more nodes not in said first network.

36. The process of claim 24 further comprising the step that if a peer to peer message is from a node in a first network and is a query message and if said query message is unsuccessful, forwarding said query message to a weighted list of nodes in said first network.

37. The process of claim 24 further comprising the step that if a peer to peer message is not from a node in a first network and is a query message, ignoring said query message.

38. The process of claim 24 further comprising the step that if a peer to peer message is a queryhit message, determining if a matching query message has been observed and if so forwarding said queryhit message to the originator

of said matching query message.

39. The process of claim 24 further comprising the step that if a peer to peer message is a query message said process determining:

a) if said query is from a node in a first network, determining if said query may be serviced by one or more nodes in said first network, if so, forwarding said query to said one or more nodes in said first network; or

b) if said query is from a node not in said first network, dropping said query.

40. The process of claim 24 further comprising the step that if a peer to peer message is a pong and a matching ping message has not been seen, dropping said pong.

41. The process of claim 24 further comprising the step that if a peer to peer message is a ping, optionally dropping said ping.

42. The process of claim 41 further comprising the step that if said ping has not been dropped, creating one or more pong messages to nodes in the same cost class as the node sending said ping.

43. The process of claim 24 further comprising the step that if a peer to peer message is a connect, and if said connect is better than an existing connection, dropping said existing connection for said connect.

44. The process of claim 24 further comprising the step that said cost class is set by an administrator.

45. The process of claim 24 further comprising the step that said cost class is set by said process.

46. The process of claim 29 further comprising the step that if no nodes are found in said content index to satisfy a peer to peer query message, forwarding said query message to a set of nodes determined by a weighted probabilistic function.

47. A computer program (116) comprising code means adapted to perform all the steps of any of claims 24 to 46, wherein said program is run on a data processing system.

48. The computer program as claimed in claim 47 embodied on a computer readable medium.

Patentansprüche

1. Peer-zu-Peer-Verkehrsoptimierer (10), umfassend:

Mittel zum Überwachen aller Peer-zu-Peer-Nachrichten zwischen und innerhalb von Netzwerken (12), die mit dem Optimierer verbunden sind;
Mittel zum Pflegen von Kostenklasseninformationen jedes Peer-zu-Peer-Knotens (14);
Mittel zum Zuweisen von Pfadkosten zu jeder vorgeschlagenen Verbindung auf der Grundlage der Kostenklasse jedes Knotens (14) in der Verbindung;
Mittel zum Nutzen der Pfadkosten, um einen kosteneffizienten Pfad für die vorgeschlagene Verbindung zu bestimmen; und
Mittel zum Erzeugen, Vernichten, Modifizieren oder Ignorieren von Peer-zu-Peer-Nachrichten, so dass die vorgeschlagene Verbindung den bestimmten kosteneffizienten Pfad nutzt.

2. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht anfangs nicht auf den kosteneffizienten Pfad geleitet wird, die Peer-zu-Peer-Nachricht auf den kosteneffizienten Pfad umgeleitet wird.

3. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht nicht von einem Knoten in einem ersten Netzwerk stammt und eine Anfragenachricht ist, die Anfragenachricht zu einem oder mehreren Knoten in nicht dem ersten Netzwerk weitergeleitet wird.

4. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht von einem Knoten in einem ersten

Netzwerk stammt und eine Anfragenachricht ist, und wenn die Anfragenachricht erfolglos ist, die Anfragenachricht zu einer gewichteten Liste von Knoten im ersten Netzwerk weitergeleitet wird.

5. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht nicht von einem Knoten in einem ersten Netzwerk stammt und eine Anfragenachricht ist, die Anfragenachricht ignoriert wird.

6. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht eine Anfragetreffernachricht ist, bestimmt wird, ob eine passende Anfragenachricht beobachtet worden ist, wobei dann, wenn dies zutrifft, die Anfragetreffernachricht zum Urheber der passenden Anfragenachricht weitergeleitet wird.

7. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht eine Anfragenachricht ist, der Optimierer:

   a) wenn die Anfrage von einem Knoten in einem ersten Netzwerk stammt, bestimmt, ob die Anfrage von einem oder mehreren Knoten im ersten Netzwerk bedient werden kann, und dann, wenn dies zutrifft, die Anfrage zu dem einen oder den mehreren Knoten im ersten Netzwerk weiterleitet; oder
   b) wenn die Anfrage nicht von einem Knoten im ersten Netzwerk stammt, die Anfrage verwirft.

8. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht ein Pong ist und keine passende Ping-Nachricht beobachtet worden ist, der Pong verworfen wird.

9. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht ein Ping ist, der Optimierer optional den Ping verwirft.

10. Optimierer nach Anspruch 9, bei dem dann, wenn der Optimierer den Ping nicht verworfen hat, der Optimierer eine oder mehrere Pong-Nachrichten für Knoten in der gleichen Kostenklasse wie der Knoten, der den Ping gesendet hat, erzeugt.

11. Optimierer nach Anspruch 1, bei dem dann, wenn eine Peer-zu-Peer-Nachricht eine Verbindung ist, und wenn die Verbindung besser ist als eine bestehende Verbindung, die bestehende Verbindung für diese Verbindung verworfen wird.

12. Optimierer nach Anspruch 1, bei dem die Kostenklasse von einem Administrator des Optimierers festgelegt wird.

13. Optimierer nach Anspruch 1, bei dem die Kostenklasse vom Optimierer festgelegt wird.

14. Optimierer nach Anspruch 1, bei dem jede Peer-zu-Peer-Nachricht untersucht wird, um Informationen über den Urheber der Nachricht zu erhalten.

15. Optimierer nach Anspruch 14, bei dem die Informationen vom Optimierer in einem Host-Pufferspeicher gespeichert werden.

16. Optimierer nach Anspruch 1, der ferner einen Inhaltsindex umfasst, wobei der Inhaltsindex eine Abbildung des Inhalts auf die Knoten umfasst.

17. Optimierer nach Anspruch 16, bei dem der Inhaltsindex genutzt wird, um die Anzahl der Knoten zu begrenzen, zu denen eine Anfragenachricht weitergeleitet wird.

18. Optimierer nach Anspruch 17, bei dem der Inhaltsindex benutzt wird, um einen optimierten Satz von Knoten zum Empfangen der Anfragenachricht zu bestimmen, wobei der optimierte Satz von Knoten auf den Pfadkosten beruht.

19. Optimierer nach Anspruch 18, bei dem dann, wenn keine Knoten im Inhaltsindex gefunden werden, die die Anfragenachricht befriedigen, die Anfragenachricht zu einem Satz von Knoten weitergeleitet wird, der durch eine gewichtete wahrscheinlichkeitstheoretische Funktion bestimmt wird.

20. Optimierer nach Anspruch 1, bei dem das Mittel zum Erzeugen, Vernichten, Modifizieren oder Ignorieren Mittel zum Erzeugen umfasst.

21. Optimierer nach Anspruch 1, bei dem das Mittel zum Erzeugen, Vernichten, Modifizieren oder Ignorieren Mittel zum Vernichten umfasst.

22. Optimierer nach Anspruch 1, bei dem das Mittel zum Erzeugen, Vernichten, Modifizieren oder Ignorieren Mittel zum Modifizieren umfasst.

23. Optimierer nach Anspruch 1, bei dem das Mittel zum Erzeugen, Vernichten, Modifizieren oder Ignorieren Mittel zum Ignorieren umfasst.

24. Peer-zu-Peer-Verkehrsoptimierungsprozess (10), der die Schritte umfasst:

Überwachen aller Peer-zu-Peer-Nachrichten zwischen und innerhalb von Netzwerken (12);
Pflegen von Kostenklasseninformationen jedes Peer-zu-Peer-Knotens (14);
Zuweisen von Pfadkosten zu jeder vorgeschlagenen Verbindung auf der Grundlage der Kostenklasse jedes Knotens (14) in der Verbindung;
Nutzen der Pfadkosten, um einen kosteneffizienten Pfad für die vorgeschlagene Verbindung zu bestimmen; und
Erzeugen, Vernichten, Modifizieren oder Ignorieren von Peer-zu-Peer-Nachrichten, so dass die vorgeschlagene Verbindung den bestimmten Pfad verwendet.

25. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht anfangs nicht auf den kosteneffizienten Pfad geleitet wird, die Peer-zu-Peer-Nachricht auf den kosteneffizienten Pfad umleitet.

26. Prozess nach Anspruch 24, der ferner einen Schritt des Nutzens einer Nachricht zum Erhalten von Informationen über den Urheber der Nachricht umfasst.

27. Prozess nach Anspruch 26, der ferner den Schritt des Speicherns der Informationen in einem Host-Pufferspeicher umfasst.

28. Prozess nach Anspruch 24, der ferner einen Schritt des Abgleichens eines angeforderten Dateinamens mit in einem Inhaltsindex gespeicherten Dateinamen umfasst.

29. Prozess nach Anspruch 28, der ferner einen Schritt des Nutzens des Inhaltsindex bei der Bestimmung des kosteneffizienten Pfades umfasst.

30. Prozess nach Anspruch 24, bei dem der Schritt des Erzeugens, Vernichtens, Modifizierens oder Ignorierens das Modifizieren umfasst.

31. Prozess nach Anspruch 24, der ferner einen Schritt des Weiterleitens einer Peer-zu-Peer-Nachricht umfasst.

32. Prozess nach Anspruch 24, bei dem der Schritt des Erzeugens, Vernichtens, Modifizierens oder Ignorierens das Ignorieren umfasst.

33. Prozess nach Anspruch 24, bei dem der Schritt des Erzeugens, Vernichtens, Modifizierens oder Ignorierens das Erzeugen umfasst.

34. Prozess nach Anspruch 24, bei dem der Schritt des Erzeugens, Vernichtens, Modifizierens oder Ignorierens das Vernichten umfasst.

35. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht nicht von einem Knoten in einem ersten Netzwerk stammt und eine Anfragenachricht ist, die Anfragenachricht zu einem oder mehreren Knoten in nicht dem ersten Netzwerk weiterleitet.

36. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht von einem Knoten in einem ersten Netzwerk stammt und eine Anfragenachricht ist, und wenn die Anfragenachricht erfolglos ist, die Anfragenachricht zu einer gewichteten Liste von Knoten im ersten Netzwerk weiterleitet.

37. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht nicht von einem Knoten in einem ersten Netzwerk stammtund eine Anfragenachricht ist, die Anfragenachricht ignoriert.

38. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht eine Anfragetreffernachricht ist, bestimmt, ob eine passende Anfragenachricht beobachtet worden ist, und dann, wenn dies zutrifft, die Anfragetreffernachricht zum Urheber der passenden Anfragenachricht weiterleitet.

39. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht eine Anfragenachricht ist und:

a) wenn eine Anfrage von einem Knoten in einem ersten Netzwerk stammt, bestimmt, ob die Anfrage von einem oder mehreren Knoten im ersten Netzwerk bedient werden kann, und dann, wenn dies zutrifft, die Anfrage zu dem einen oder den mehreren Knoten im ersten Netzwerk weiterleitet; oder
b) wenn die Anfrage nicht von einem Knoten im ersten Netzwerk stammt, die Anfrage verwirft.

40. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht ein Pong ist und keine passende Ping-Nachricht beobachtet worden ist, den Pong verwirft.

41. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht ein Ping ist, optional den Ping verwirft.

42. Prozess nach Anspruch 41, der ferner einen Schritt umfasst, der dann, wenn der Ping nicht verworfen worden ist, eine oder mehrere Pong-Nachrichten für Knoten in der gleichen Kostenklasse wie der Knoten, der den Ping gesendet hat, erzeugt.

43. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, der dann, wenn eine Peer-zu-Peer-Nachricht eine Verbindung ist, und wenn die Verbindung besser ist als eine bestehende Verbindung, die bestehende Verbindung für diese Verbindung verwirft.

44. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, bei dem die Kostenklasse von einem Administrator festgelegt wird.

45. Prozess nach Anspruch 24, der ferner einen Schritt umfasst, bei dem die Kostenklasse durch den Prozess festgelegt wird.

46. Prozess nach Anspruch 29, der ferner einen Schritt umfasst, der dann, wenn kein Knoten im Inhaltsindex gefunden wird, der eine Peer-zu-Peer-Anfragenachricht befriedigt, die Anfragenachricht zu einem Satz von Knoten weiterleitet, der durch eine gewichtete wahrscheinlichkeitstheoretische Funktion bestimmt wird.

47. Computerprogramm (116), das Codemittel umfasst, die dafür ausgelegt sind, alle Schritte von irgendeinem der Ansprüche 24 bis 46 auszuführen, wobei das Programm auf einem Datenverarbeitungssystem läuft.

48. Computerprogramm nach Anspruch 47, das auf einem computerlesbaren Medium ausgeführt ist.

Revendications

1. Dispositif d'optimisation du trafic poste à poste (10), ledit dispositif d'optimisation comprenant :

un moyen pour surveiller tous les messages poste à poste entre des réseaux et à l'intérieur de réseaux (12) connectés audit dispositif d'optimisation ;
un moyen pour maintenir une information de classe de coût de chaque noeud poste à poste (14) ;
un moyen pour affecter un coût de chemin à chaque connexion proposée sur la base de la classe de coût de chaque noeud (14) dans la connexion ;
un moyen pour utiliser ledit coût de chemin pour déterminer un chemin pertinent en termes de coût pour ladite connexion proposée ; et
un moyen pour créer, détruire, modifier ou ignorer des messages poste à poste de sorte que ladite connexion proposée utilise ledit chemin pertinent en termes de coût.

2. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste n'est pas acheminé initialement sur ledit chemin pertinent en termes de coût, ledit message poste à poste est réacheminé sur ledit chemin pertinent en termes de coût.

3. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste ne provient pas d'un noeud dans un premier réseau et si c'est un message de requête, ledit message de requête est acheminé à un ou plusieurs noeuds ne se trouvant pas dans ledit premier réseau.

4. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste provient d'un noeud dans un premier réseau et si c'est un message de requête et si ledit message de requête est infructueux, ledit message de requête est acheminé à une liste pondérée de noeuds dans ledit premier réseau.

5. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste ne provient pas d'un noeud dans un premier réseau et si c'est un message de requête, ledit message de requête est ignoré.

6. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste est un message de type Queryhit, il est déterminé si un message de requête correspondant a été observé et, si c'est le cas, ledit message de type Queryhit est acheminé à l'expéditeur dudit message de requête correspondant.

7. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste est un message de requête, ledit dispositif d'optimisation détermine :

   a) si le message de requête provient d'un noeud dans un premier réseau, il est déterminé si ladite requête peut être prise en charge par un ou plusieurs noeuds dans ledit premier réseau et, si c'est le cas, ladite requête est acheminée auxdits un ou plusieurs noeuds dans ledit premier réseau ; et
   b) si le message de requête ne provient pas d'un noeud dans ledit premier réseau, ladite requête est abandonnée.

8. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste est un message pong et si un message ping correspondant n'a pas été vu, ledit message pong est abandonné.

9. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste est un message ping, ledit dispositif d'optimisation abandonne ledit message ping.

10. Dispositif d'optimisation selon la revendication 9, dans lequel, si ledit dispositif d'optimisation n'a pas abandonné ledit message ping, ledit dispositif d'optimisation crée un ou plusieurs messages pong sur les noeuds dans la même classe de coût que le noeud qui envoie ledit message ping.

11. Dispositif d'optimisation selon la revendication 1, dans lequel, si un message poste à poste est une connexion et si ladite connexion est meilleure qu'une connexion existante, ladite connexion existante est abandonnée au profit de ladite connexion.

12. Dispositif d'optimisation selon la revendication 1, dans lequel ladite classe de coût est réglée par un administrateur dudit dispositif d'optimisation.

13. Dispositif d'optimisation selon la revendication 1, dans lequel ladite classe de coût est réglée par ledit dispositif d'optimisation.

14. Dispositif d'optimisation selon la revendication 1, dans lequel chaque message poste à poste est examiné pour obtenir une information sur l'expéditeur dudit message.

15. Dispositif d'optimisation selon la revendication 14, dans lequel ladite information est stockée par ledit dispositif d'optimisation dans une antémémoire centrale.

16. Dispositif d'optimisation selon la revendication 1, comprenant de plus un index de contenu, ledit index de contenu comprenant une mise en correspondance du contenu avec des noeuds.

17. Dispositif d'optimisation selon la revendication 16, dans lequel ledit index de contenu est utilisé pour limiter le

nombre de noeuds sur lesquels est acheminé un message de requête.

18. Dispositif d'optimisation selon la revendication 17, dans lequel ledit index de contenu est utilisé pour déterminer un ensemble optimisé de noeuds pour recevoir ledit message de requête, ledit ensemble optimisé de noeuds étant basé sur le coût de chemin.

19. Dispositif d'optimisation selon la revendication 18, dans lequel, si aucun noeud n'est trouvé dans ledit index de contenu pour satisfaire audit message de requête, ledit message de requête est acheminé à un ensemble de noeuds déterminé par une fonction probabiliste pondérée.

20. Dispositif d'optimisation selon la revendication 1, dans lequel le moyen pour créer, détruire, modifier ou ignorer comprend un moyen pour créer.

21. Dispositif d'optimisation selon la revendication 1, dans lequel le moyen pour créer, détruire, modifier ou ignorer comprend un moyen pour détruire.

22. Dispositif d'optimisation selon la revendication 1, dans lequel le moyen pour créer, détruire, modifier ou ignorer comprend un moyen pour modifier.

23. Dispositif d'optimisation selon la revendication 1, dans lequel le moyen pour créer, détruire, modifier ou ignorer comprend un moyen pour ignorer.

24. Procédé d'optimisation du trafic poste à poste (10), ledit procédé comprenant les étapes qui consistent à :

surveiller tous les messages poste à poste entre des réseaux et à l'intérieur de réseaux (12) ;
maintenir une information de classe de coût de chaque noeud poste à poste (14) ;
affecter un coût de chemin à chaque connexion proposée sur la base de ladite classe de coût de chaque noeud (14) dans la connexion ;
utiliser ledit coût de chemin pour déterminer un chemin pertinent en termes de coût pour ladite connexion proposée ; et
créer, détruire, modifier ou ignorer des messages poste à poste de sorte que ladite connexion proposée utilise ledit chemin déterminé.

25. Procédé selon la revendication 24, comprenant de plus l'étape qui consiste, si un message poste à poste n'est pas acheminé initialement sur ledit chemin pertinent en termes de coût, à réacheminer ledit message poste à poste sur ledit chemin pertinent en termes de coût.

26. Procédé selon la revendication 24, comprenant de plus l'étape qui consiste à utiliser un message pour obtenir une information sur l'expéditeur dudit message.

27. Procédé selon la revendication 26, comprenant de plus l'étape qui consiste à stocker ladite information dans une antémémoire centrale.

28. Procédé selon la revendication 24, comprenant de plus l'étape qui consiste à mettre en correspondance un nom de fichier demandé avec des noms de fichier stockés dans un index de contenu.

29. Procédé selon la revendication 28, comprenant de plus l'étape qui consiste à utiliser ledit index de contenu dans la détermination dudit chemin pertinent en termes de coût.

30. Procédé selon la revendication 24, dans lequel l'étape qui consiste à créer, détruire, modifier ou ignorer comprend la modification.

31. Procédé selon la revendication 24 comprenant de plus l'étape qui consiste à acheminer un message poste à poste.

32. Procédé selon la revendication 24, dans lequel l'étape qui consiste à créer, détruire, modifier ou ignorer comprend l'ignorance.

33. Procédé selon la revendication 24, dans lequel l'étape qui consiste à créer, détruire, modifier ou ignorer comprend

la création.

34. Procédé selon la revendication 24, dans lequel l'étape qui consiste à créer, détruire, modifier ou ignorer comprend la destruction.

35. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste ne provient pas d'un noeud dans un premier réseau et si c'est un message de requête, consiste à acheminer ledit message de requête à un ou plusieurs noeuds ne se trouvant pas dans ledit premier réseau.

36. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste provient d'un noeud dans un premier réseau et si c'est un message de requête et si ledit message de requête est infructueux, consiste à acheminer ledit message de requête à une liste pondérée de noeuds dans ledit premier réseau.

37. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste ne provient pas d'un noeud dans un premier réseau et si c'est un message de requête, consiste à ignorer ledit message de requête.

38. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste est un message de type Queryhit, consiste à déterminer si un message de requête correspondant a été observé et, si c'est le cas, à acheminer ledit message de type Queryhit à l'expéditeur dudit message de requête correspondant.

39. Procédé selon la revendication 24, comprenant de plus l'étape dans laquelle, si un message poste à poste est un message de requête, ledit procédé détermine :

   a) si le message de requête provient d'un noeud dans un premier réseau, détermine si ladite requête peut être prise en charge par un ou plusieurs noeuds dans ledit premier réseau et, si c'est le cas, ladite requête est acheminée auxdits un ou plusieurs noeuds dans ledit premier réseau ; ou
   b) si le message de requête ne provient pas d'un noeud dans un premier réseau, ladite requête est abandonnée.

40. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste est un message pong et si un message ping correspondant n'a pas été vu, consiste à abandonner ledit message pong.

41. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste est un message ping, consiste à abandonner ledit message ping de façon optionnelle.

42. Procédé selon la revendication 41, comprenant de plus l'étape qui, si ledit message ping n'a pas été abandonné, consiste à créer un ou plusieurs messages pong sur les noeuds dans la même classe de coût que le noeud qui envoie ledit message ping.

43. Procédé selon la revendication 24, comprenant de plus l'étape qui, si un message poste à poste est une connexion et si ladite connexion est meilleure qu'une connexion existante, consiste à abandonner ladite connexion existante au profit de ladite connexion.

44. Procédé selon la revendication 24, comprenant de plus l'étape qui consiste à régler ladite classe de coût par un administrateur.

45. Procédé selon la revendication 24, comprenant de plus l'étape qui consiste à régler ladite classe de coût par ledit procédé.

46. Procédé selon la revendication 29, comprenant de plus l'étape qui, si aucun noeud n'est trouvé dans ledit index de contenu pour satisfaire audit message de requête poste à poste, consiste à acheminer ledit message de requête à un ensemble de noeuds déterminé par une fonction probabiliste pondérée.

47. Programme informatique (116) comprenant un moyen de codage adapté pour exécuter toutes les étapes de l'une quelconque des revendications 24 à 46, dans lequel ledit programme tourne sur un système de traitement de données.

48. Programme informatique selon la revendication 47 réalisé sur un support lisible par ordinateur.

FIG. 1

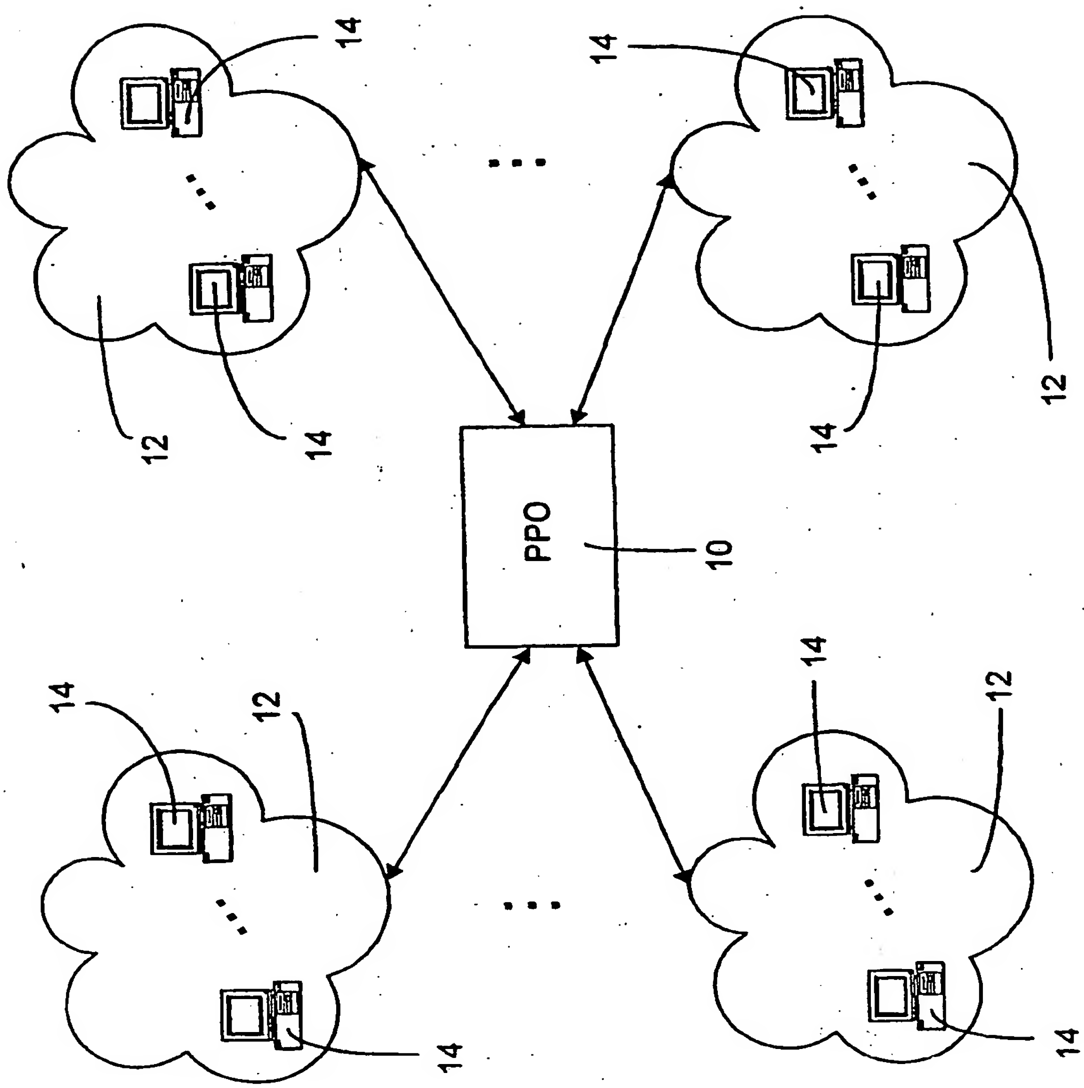
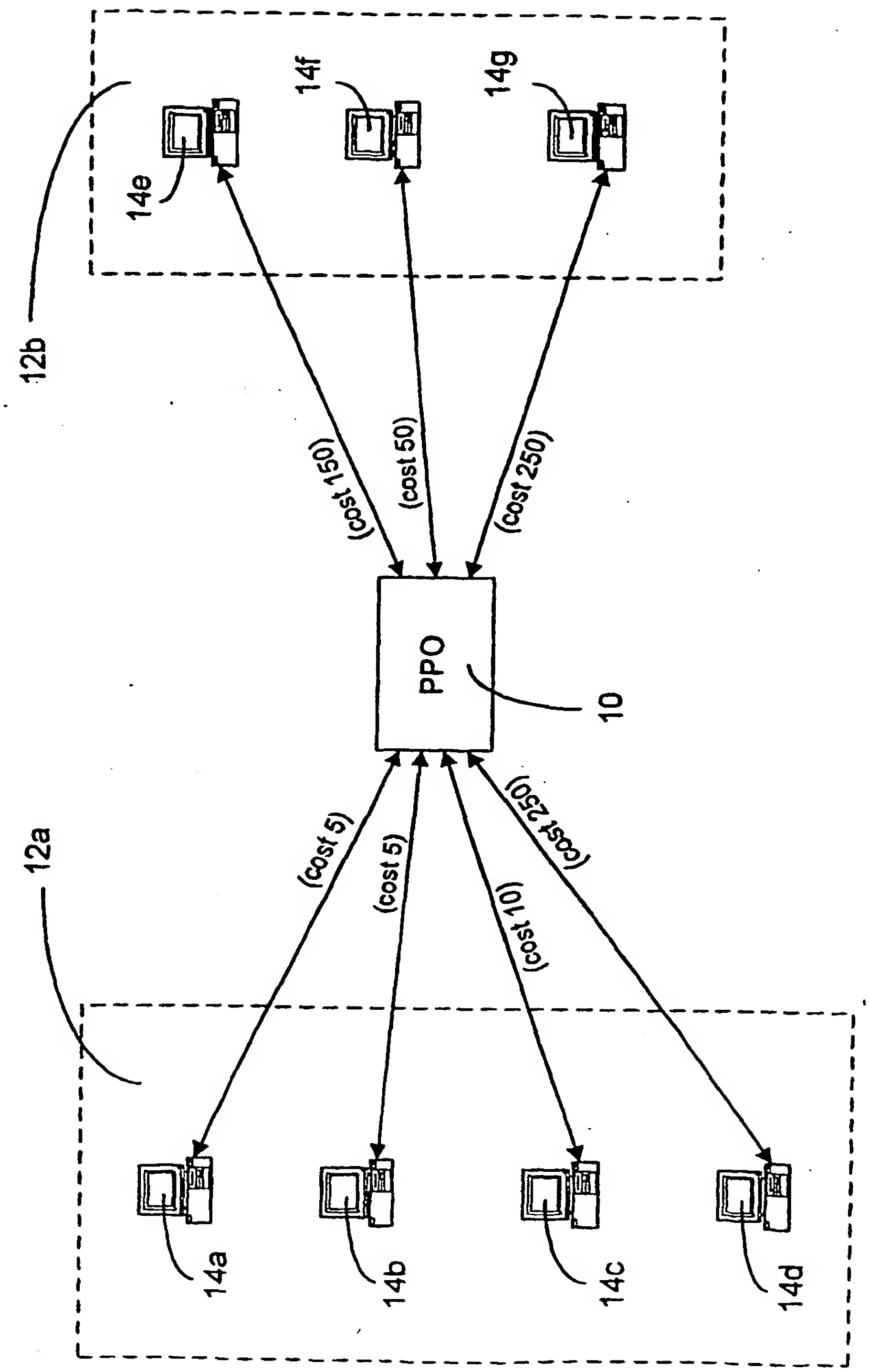
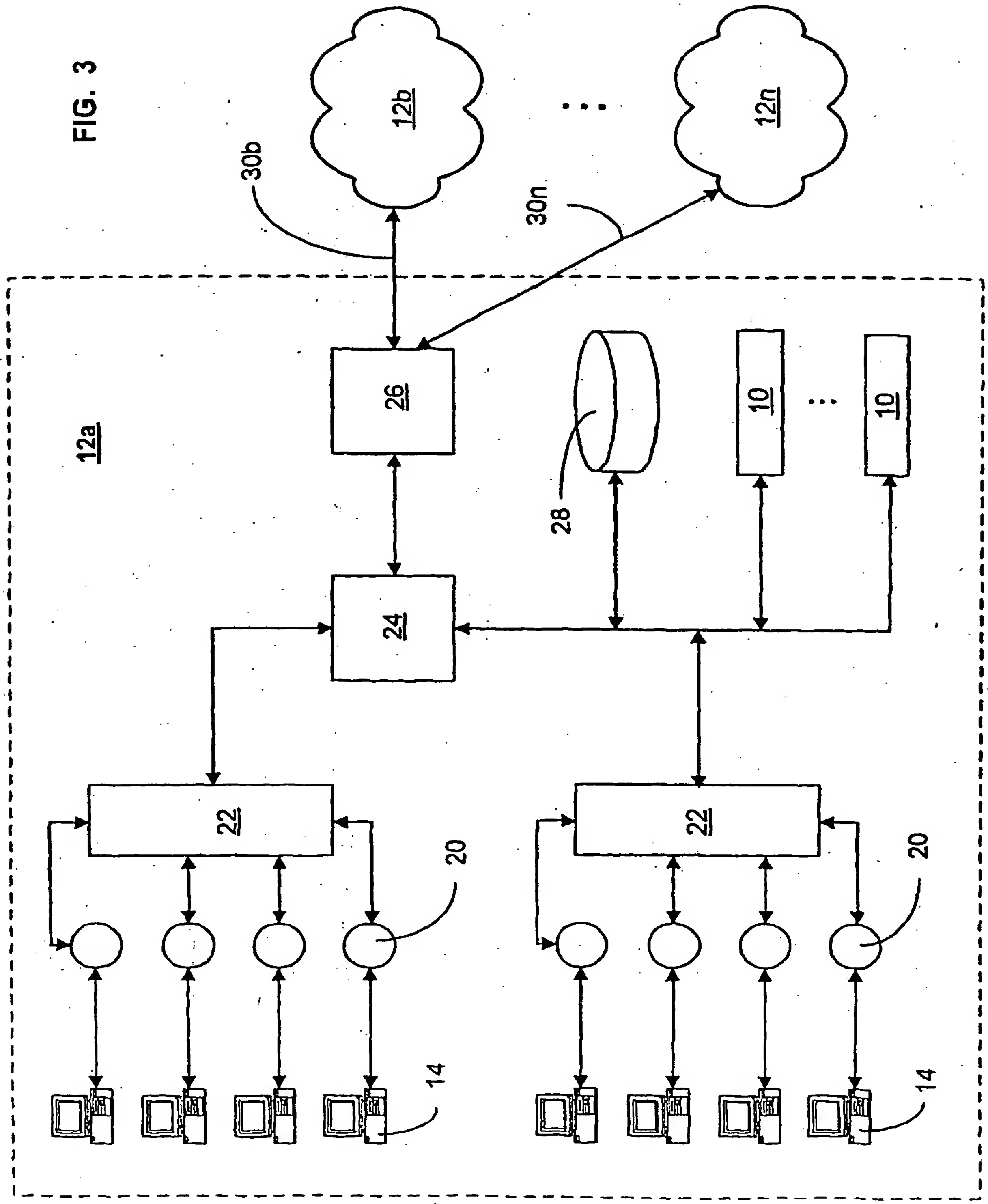


FIG. 2



12b
14e
14f
14g
(cost 150)
(cost 50)
(cost 250)
PPO
10
12a
(cost 5)
(cost 5)
(cost 10)
(cost 250)
14a
14b
14c
14d

FIG. 3



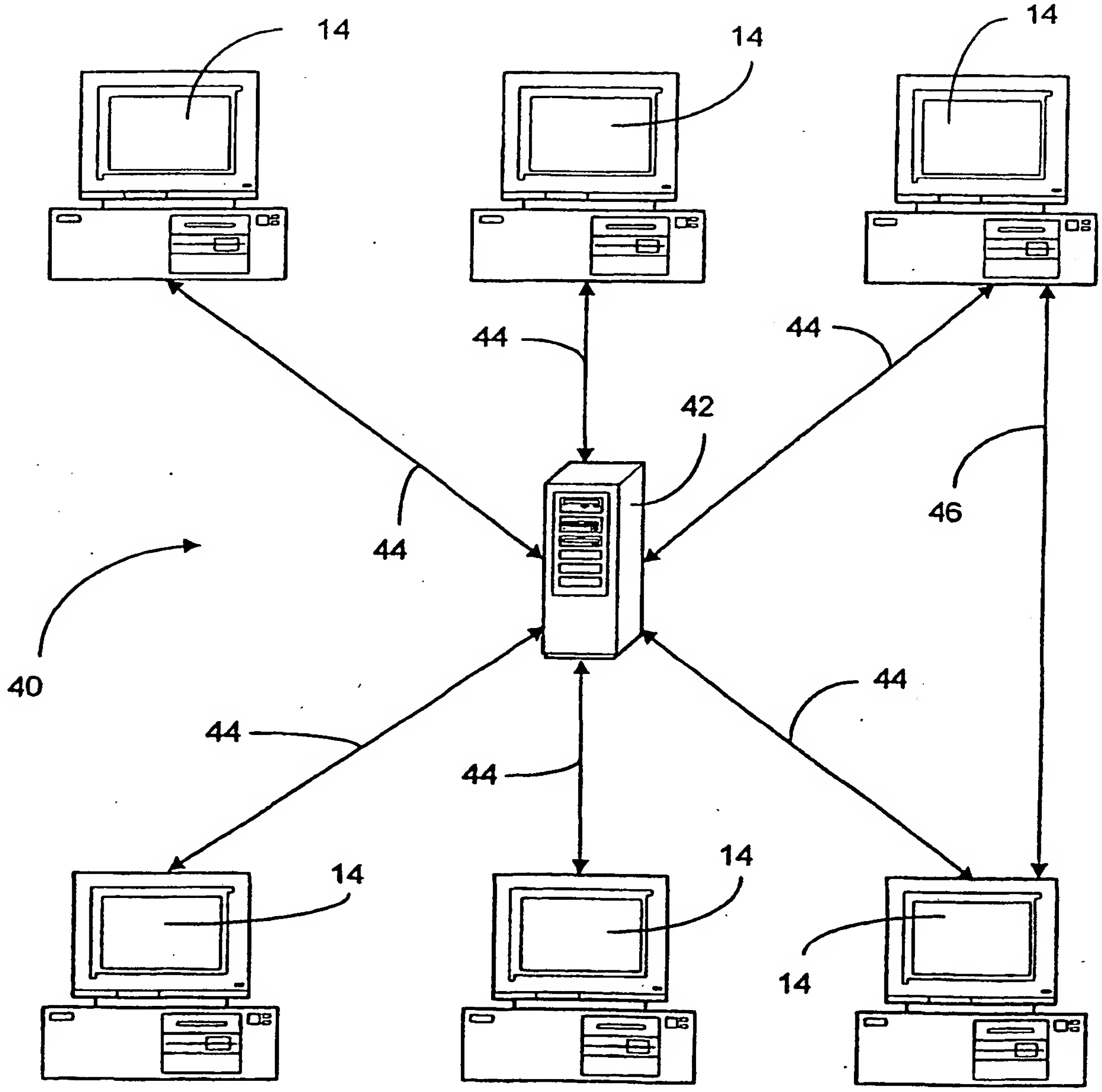


FIG. 4

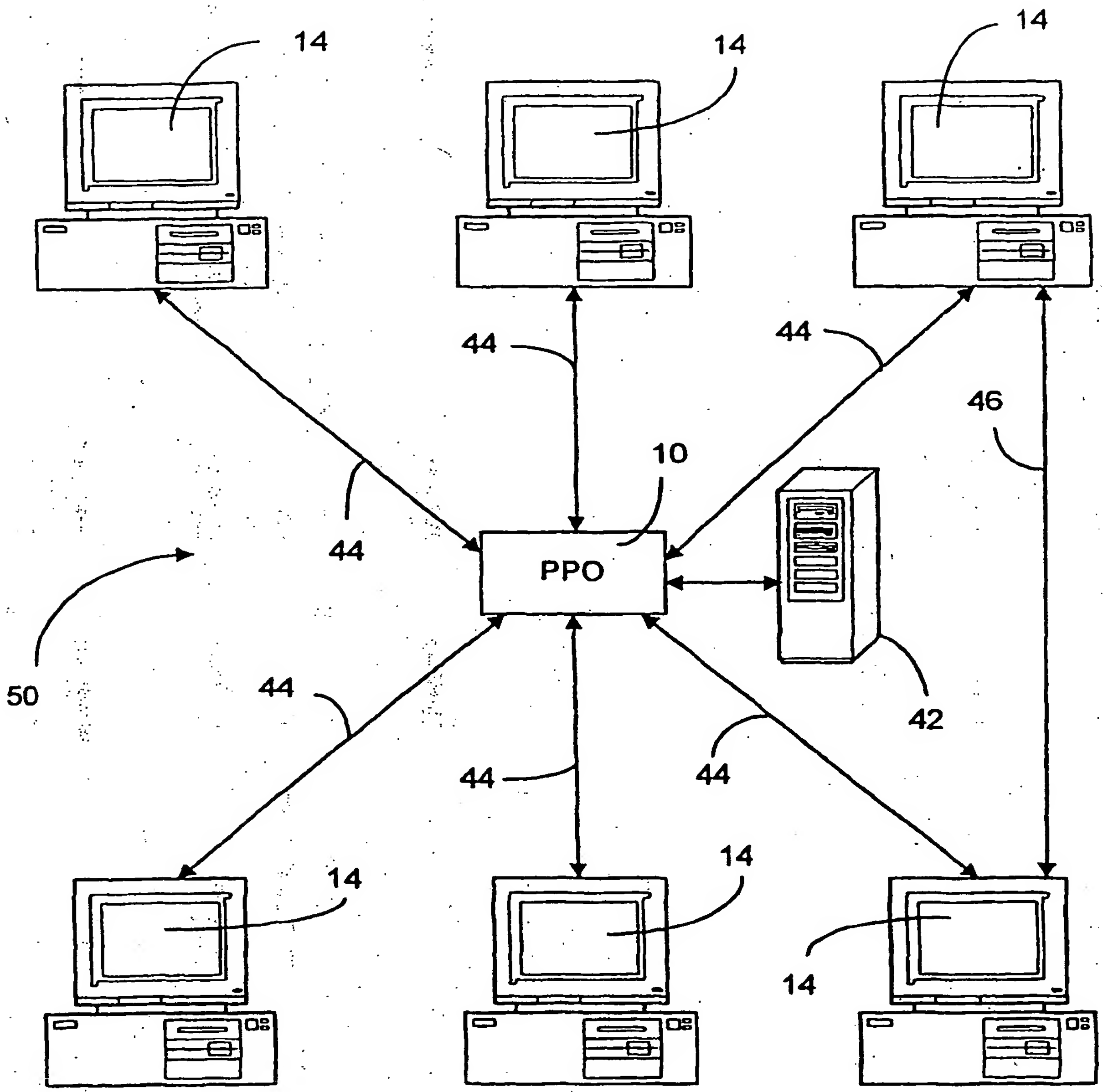


FIG. 5

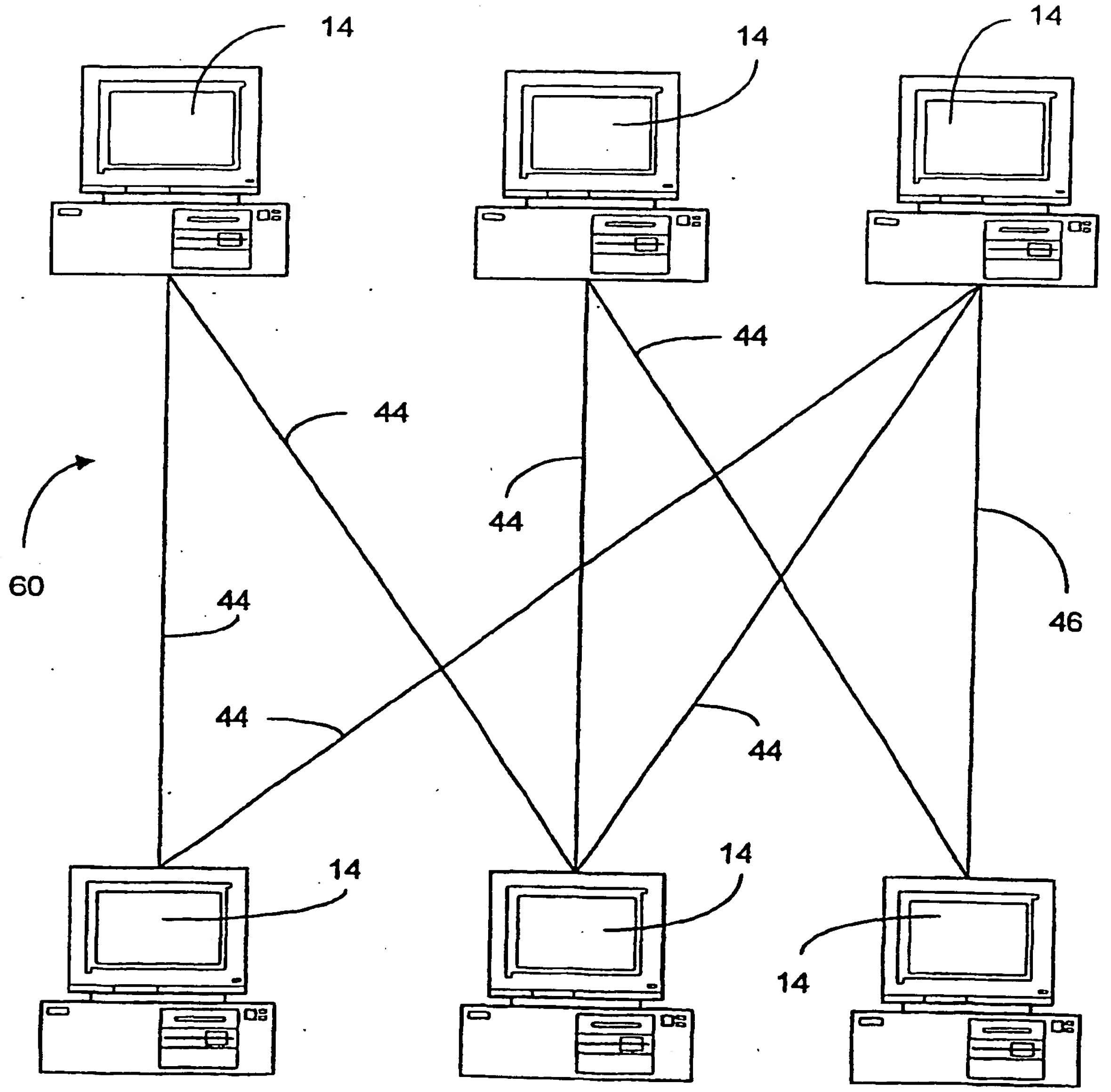


FIG. 6

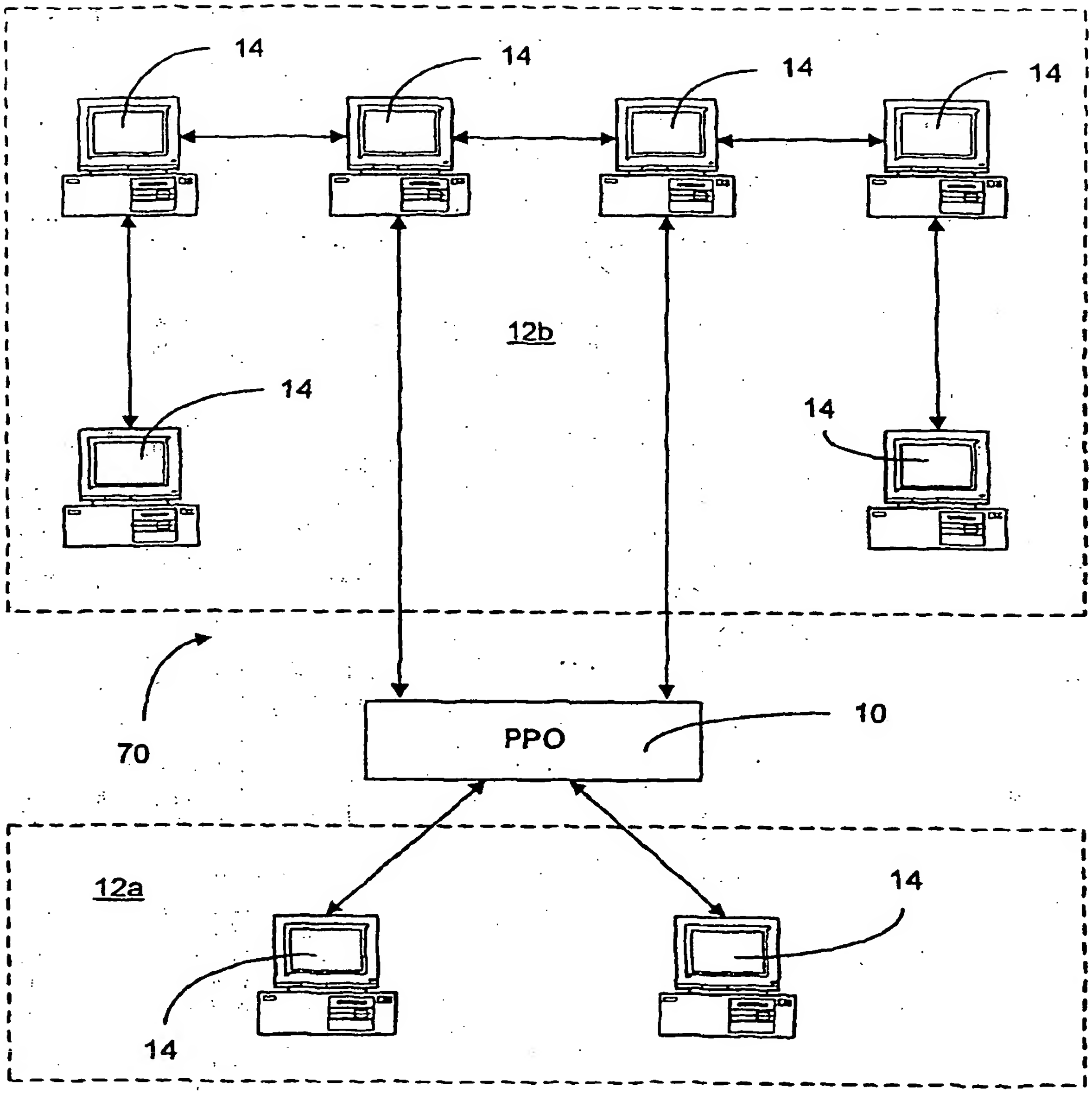


FIG. 7

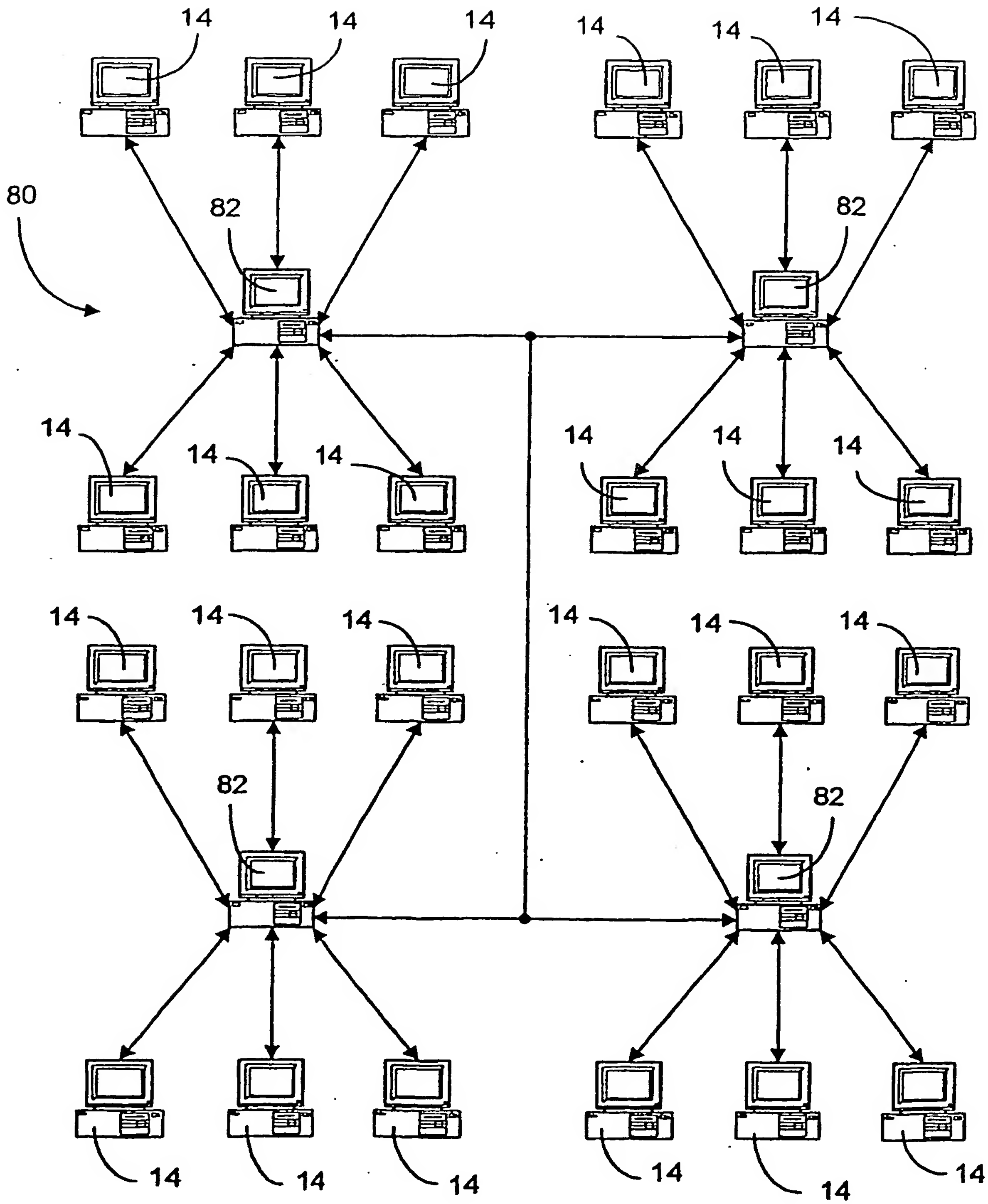

14
14
14
14
14
14
80
82
82
14
14
14
14
14
14
14
14
14
14
14
14
82
82
14
14
14
14
14
14

FIG. 8

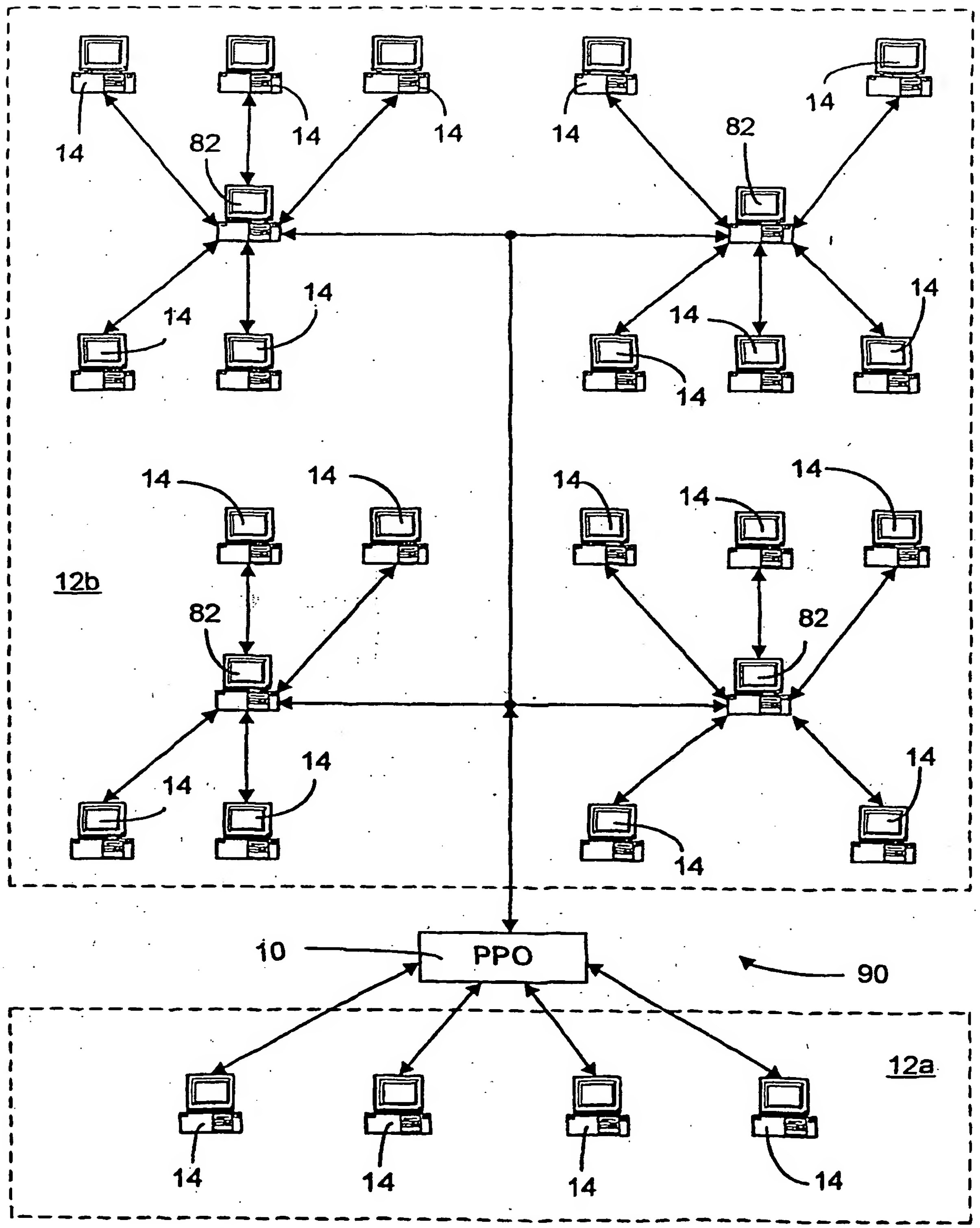

14
82
14
14
14
82
14
14
14
14
14
14
14
14
14
14
14
14
14
12b
82
82
14
14
14
14
14
PPO
10
90
12a
14
14
14
14

FIG. 9

FIG. 10

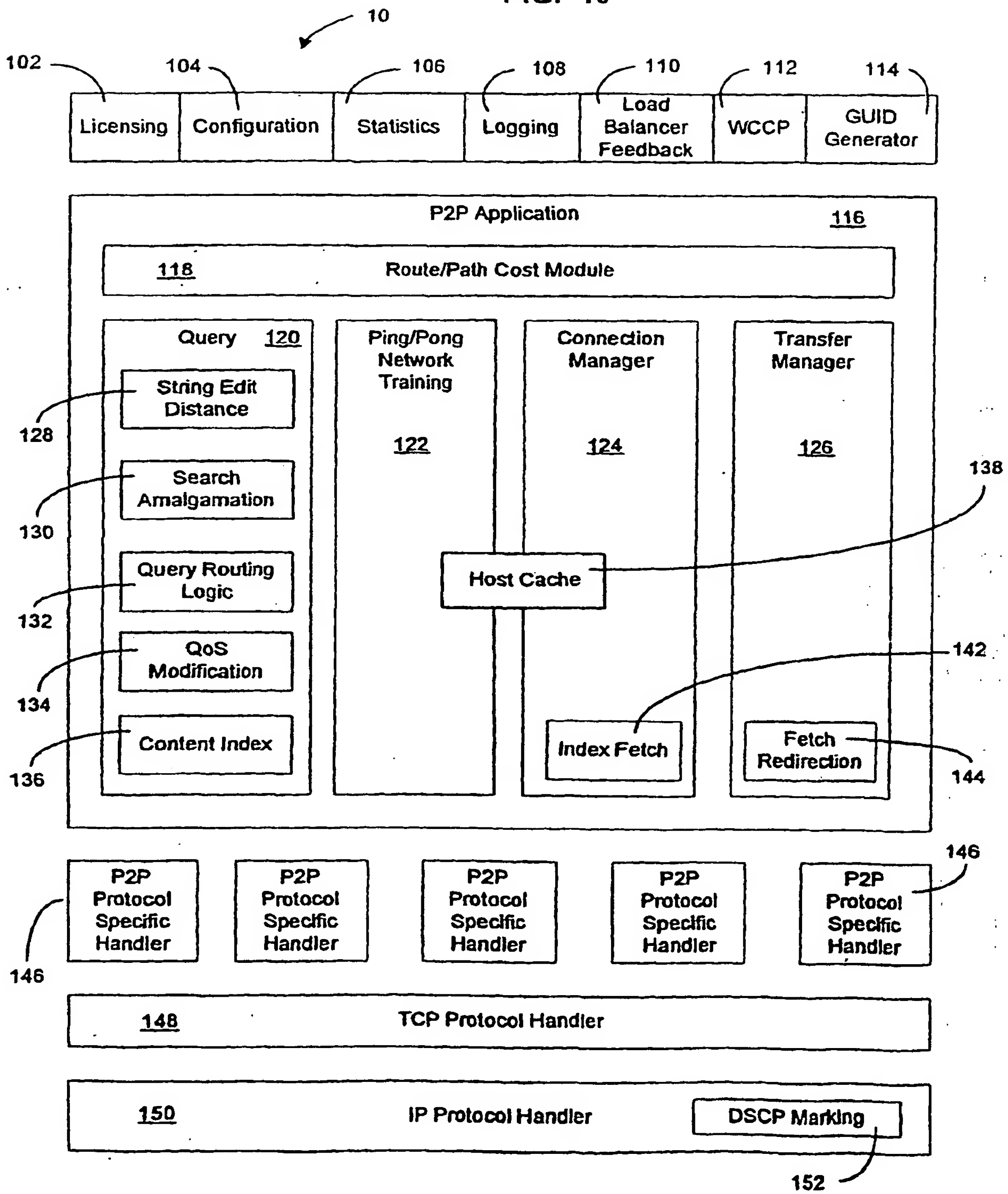

10
102
104
106
108
110
112
114
Licensing
Configuration
Statistics
Logging
Load Balancer Feedback
WCCP
GUID Generator
P2P Application
116
118
Route/Path Cost Module
Query
120
String Edit Distance
128
Search Amalgamation
130
Query Routing Logic
132
QoS Modification
134
Content Index
136
Ping/Pong Network Training
122
Connection Manager
124
Transfer Manager
126
138
Host Cache
142
Index Fetch
Fetch Redirection
144
146
P2P Protocol Specific Handler
P2P Protocol Specific Handler
P2P Protocol Specific Handler
P2P Protocol Specific Handler
P2P Protocol Specific Handler
146
148
TCP Protocol Handler
150
IP Protocol Handler
DSCP Marking
152

FIG. 11

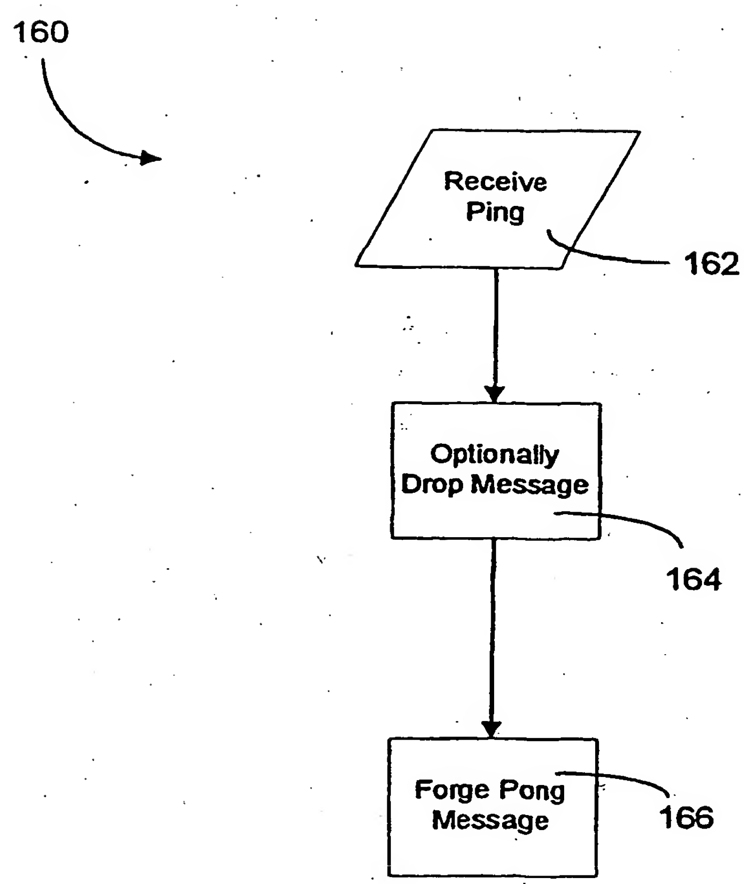

160
Receive Ping
162
Optionally Drop Message
164
Forge Pong Message
166

FIG. 12

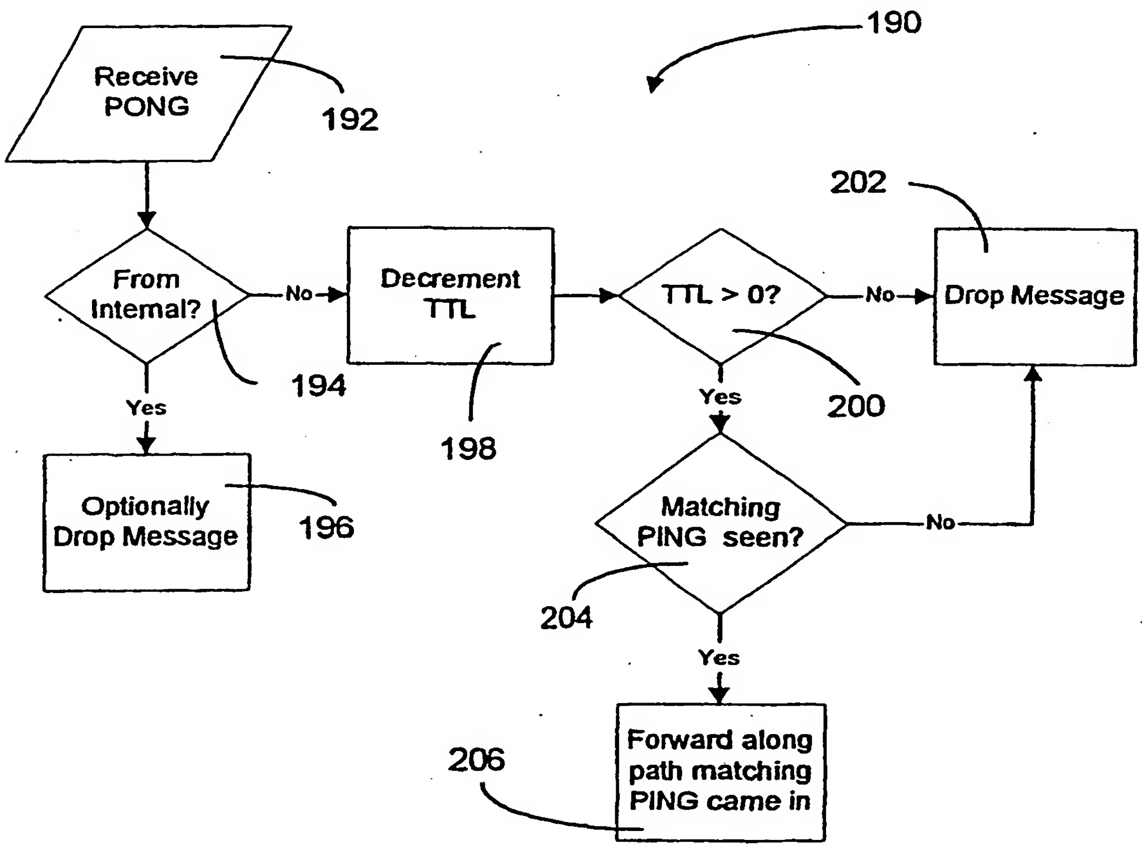

190
Receive PONG
192
From Internal?
194
No
Yes
Optionally Drop Message
196
Decrement TTL
198
TTL > 0?
200
No
Drop Message
202
Yes
Matching PING seen?
204
No
Yes
Forward along path matching PING came in
206

FIG. 13

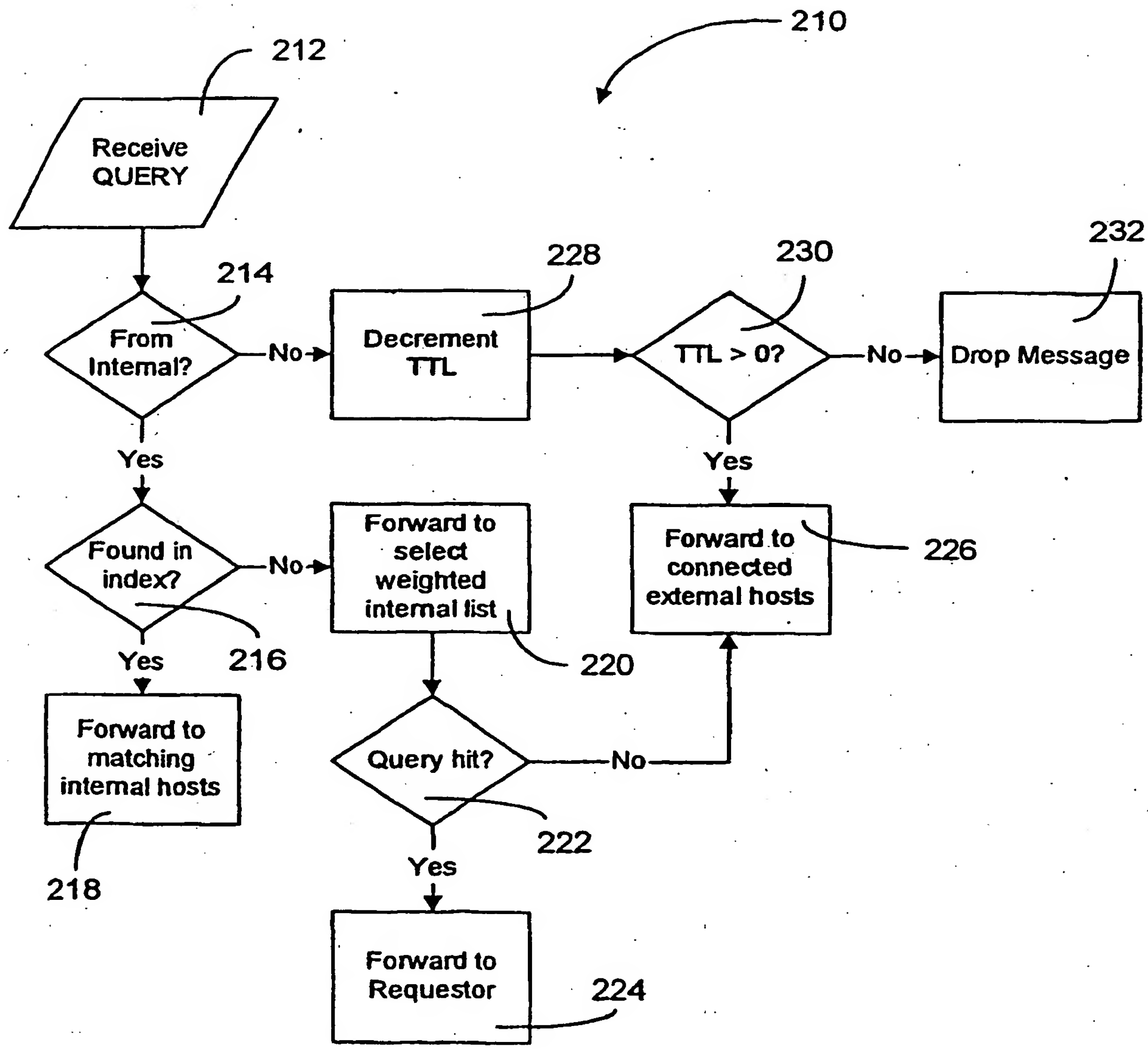
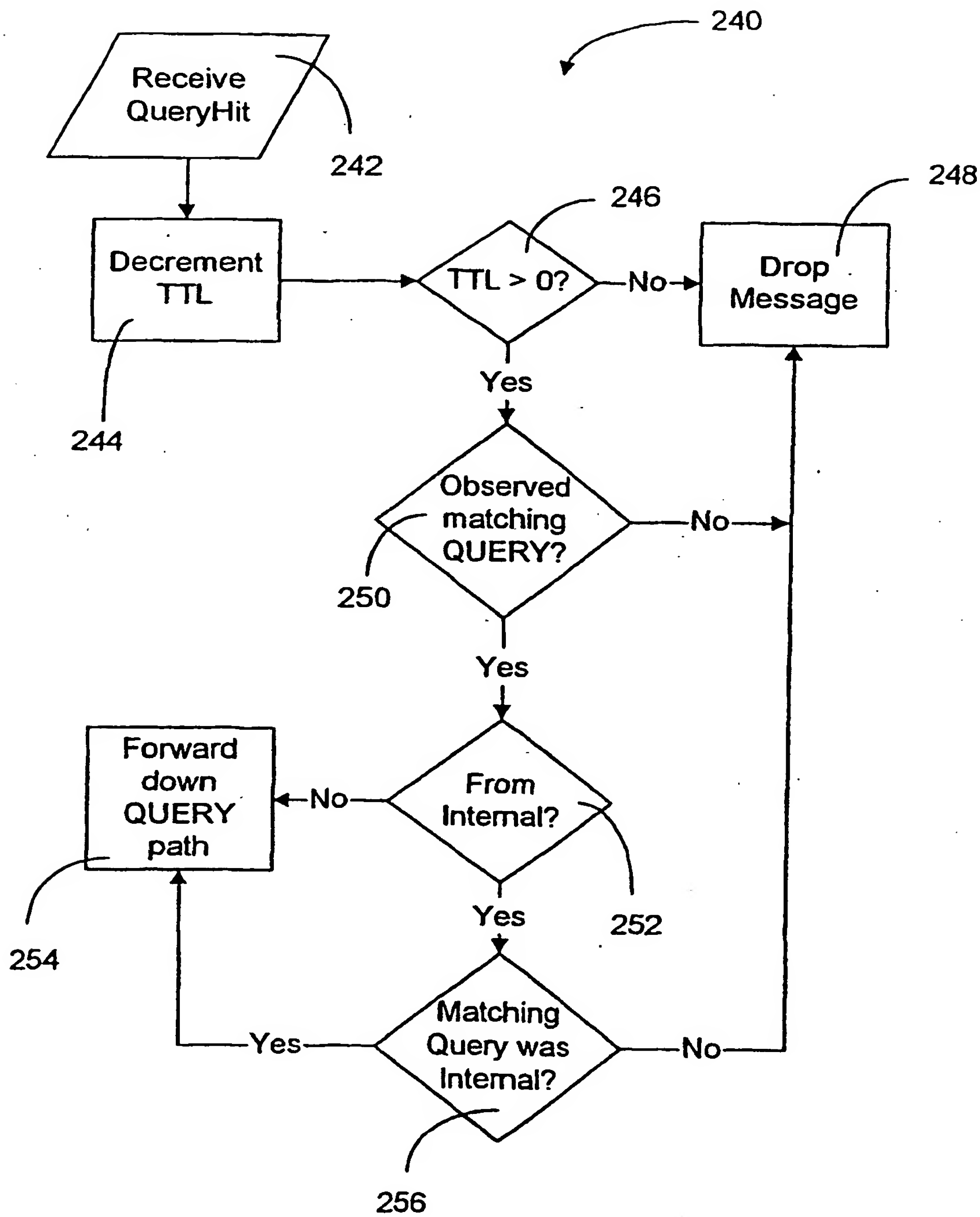
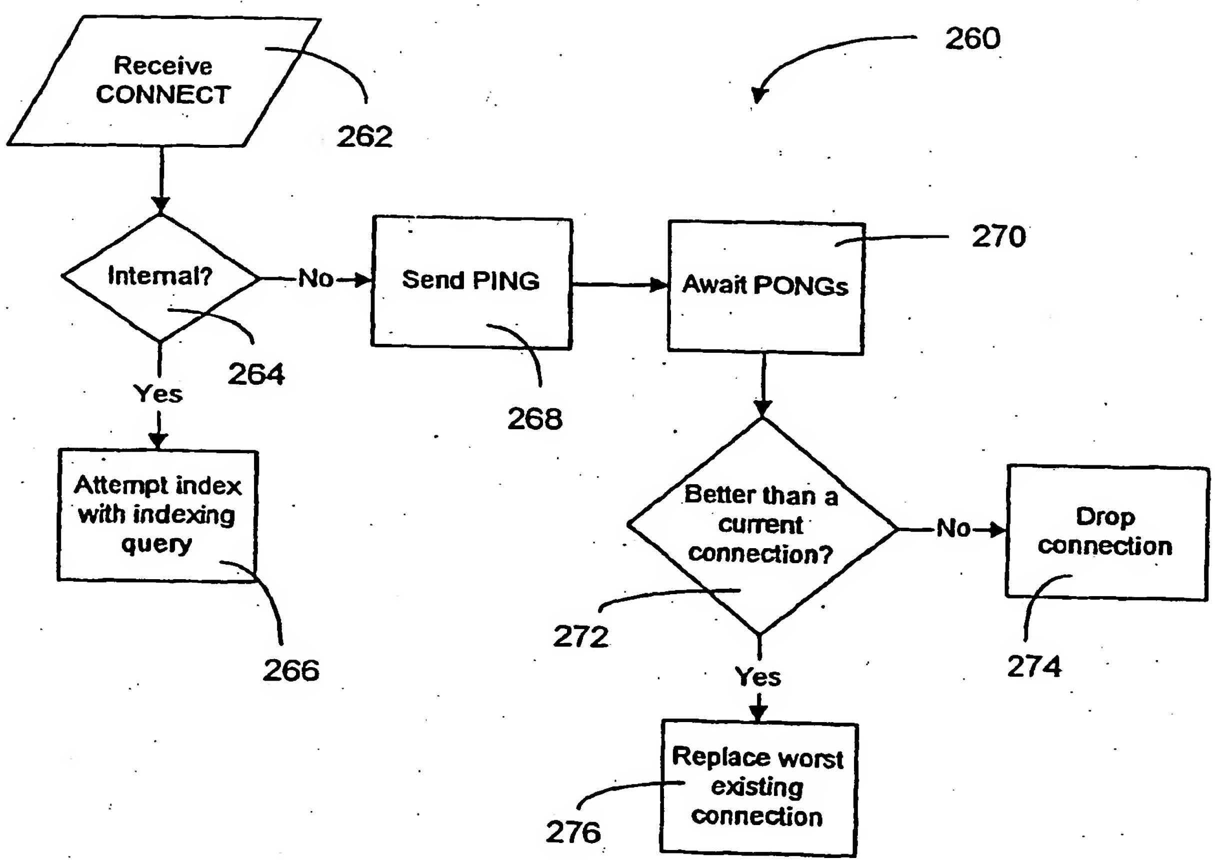

210
212
Receive QUERY
214
From Internal?
No
Decrement TTL
228
230
TTL > 0?
No
Drop Message
232
Yes
Yes
Found in index?
No
Forward to select weighted internal list
Forward to connected external hosts
226
216
Yes
220
Forward to matching internal hosts
Query hit?
No
218
222
Yes
Forward to Requestor
224

FIG. 14



240
Receive QueryHit
242
Decrement TTL
244
TTL > 0?
246
No
Drop Message
248
Yes
Observed matching QUERY?
250
No
Yes
From Internal?
252
No
Forward down QUERY path
254
Yes
Matching Query was Internal?
256
Yes
No

FIG. 15


260
Receive CONNECT
262
Internal?
No
Send PING
Await PONGs
270
264
Yes
268
Attempt index with indexing query
266
Better than a current connection?
No
Drop connection
272
274
Yes
Replace worst existing connection
276